OPTIMIZATION-BASED APPROACHES TO SAFETY-CRITICAL CONTROL WITH APPLICATIONS TO SPACE SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By:

Mark L. Mote

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Robotics,
School of Aerospace Engineering

Georgia Institute of Technology

August 2021

OPTIMIZATION-BASED APPROACHES TO SAFETY-CRITICAL CONTROL WITH APPLICATIONS TO SPACE SYSTEMS

Approved by:

Dr. Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Eric Feron, Advisor
School of Electrical and Computer Engineering
King Abdullah University of Science and Technology

Dr. Samuel Coogan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Panagiotis Tsiotras
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kerianne Hobbs
Safe Autonomy Lead
Air Force Research Laboratory

Dr. Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: July 21, 2021

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisors, Dr. Magnus Egerstedt and Dr. Eric Feron. Specifically, I would like to thank Magnus for his mentorship, his patience, and for sharing his deep insight into research and robotics. His feedback on the technical work has been invaluable, but more than that, he has instilled in me the value of understanding the meaning and the purpose behind the research. I would also like to thank Eric, who more than anyone else, has had a positive impact on my graduate school experience. I was uncertain about whether to pursue a PhD at first, and very likely would not have done so if he had not created such a fun and uniquely challenging environment. I am deeply thankful for the constant inspiration, the guidance, the candor, the fascinating conversations, and the (often undeserved) faith he put in me. Beyond teaching me how to do research, he taught me to have the audacity to search past the local optimum, to be defiant, and importantly, to avoid the cream walls.

I would also like express my gratitude to my committee members, Dr. Samuel Coogan, Dr. Kerianne Hobbs, Dr. Panagiotis Tsiotras, and Dr. Mark Costello for the valuable feedback and for taking the time to serve on my committee. I would like to thank Sam in particular for volunteering his time and expertise, and for taking on many of the roles of an advisor. He played a considerable part in improving both the quality of the work in this document, and my research skills in general. Additionally, thank you to Kerianne, a former labmate, committee member, mentor, and fellow space fanatic. Her feedback and collaboration were immensely valuable in this final year, and to much of the work in this dissertation.

I was very fortunate to have had many research experiences outside of Georgia Tech. I would like to thank Dr. Emmanuel Grolleau for having me at ISAE-ENSMA. The experience working with him on this project helped to solidify my interest in both research and control theory. Thank you to Dr. Austin Jones for his fantastic mentorship at Lincoln Laboratory. This was certainly among the most memorable of the experiences I have had in graduate school. Though the work there did not make it directly into this document, working with Austin certainly equipped me with

a skill set that would become invaluable in these remaining years. Thank you to my mentors at JPL, Dr. Amir Rahmani, Dr. Saptarshi Bandyopadhyay, Dr. Federico Rossi. Working at JPL was a dream come true, and getting the opportunity to work on a truly interesting problem with this exceptionally brilliant group of people was one of the major highlights of my graduate school experience. Thank you again to Kerianne for going the extra mile to make the experience at AFRL both fun and intellectually fruitful. Thank you to Dr. Marco Pavone for inviting me to work with his lab, and for the time and feedback on the free-flyer project. Likewise, thank you to Andrew Bylard for his dedication to making this project possible. It was a true privilege to work with these two, and the Autonomous Systems Lab as a whole. Finally, thanks to all of the great people I met at KAUST. Though the pandemic cut our time short, it was a very valuable experience nonetheless.

Thank you to Kendra Lang and to Sean Phillips for the guidance and support on the attitude control project, as well as to everyone else that I had the opportunity to interact with at Verus research, the air force, and the space force. I would like to thank Dr. Tsiotras again for his feedback and for allowing me the opportunity to work with the ASTROS lab. Additionally, thank you to Mehregan Dor, who made the hardware demo possible, and to Matthew Abate and Corbin Klett for their vital contributions to this work.

Thank you to Dr. Anant Honkan. I credit much of my success in undergrad to his guidance, and to the rigor of the transfer program. Thank you to Pablo Afman, for all the fun and experience associated with helping to build things in those early days in ESM. Thank you to Thomas Gurriet, whose work was foundational to many of the concepts in the tutorial sections, and who initially helped to spark my interest in pursuing the topic of RTA.

Thank you to the friends, labmates, and coauthors that I met at Georgia Tech, for making the experience as enjoyable and enriching as it has been: Matt, Ben, Corbin, Kevin, Raphael, Tunde, Aqib, Jose, Yousef, TK, Chris, Jack, Sid, Maria, Gennaro, Hanqing, Aayush, Louis, Philippe, Jeremy, Mohit, Gustav, Pietro, Mehregan, Alysia, Matt, Tanish, Vishnu, Thomas, Pablo, and Drew. On that note, thanks to my former roommate John, for the camaraderie from the beginning to the end of my time at Georgia Tech. Additionally, thanks to Wes, Shantonio, Mitchell, Zach, Cather-

ine, Tim, Andi, Louae, Fred, and Antoine for the friendship and the great adventures throughout these years. Finally, thank you to my family members, Ricky, Mike, Bill, and Lavina for the endless inspiration and support. In particular, I owe the greatest debt of gratitude to my dad, without whom no part of this would have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Symbols

$\wedge$	=	Conjunction
$\vee$	=	Disjunction
$\cup$	=	Set union
$\cap$	=	Set intersection
$:=$	=	Equal to by definition
$[\underline{x}, \overline{x}]$	=	Hyperrectangular set defined by the corner points $\underline{x}$ and $\overline{x}$
$\|\cdot\|_p$	=	p-norm of a vector
$\setminus$	=	Set difference
$\subset$	=	Strict subset
$\subseteq$	=	Subset
$\varphi(x)$	=	Safety constraint function
$\phi^u(t; x)$	=	Flow evaluated t units of time forward from state x under a control law u
$\mathbb{R}$	=	Set of real numbers
$\mathbb{R}^n$	=	Set of real vectors of length n
$\mathbb{R}^{m \times n}$	=	Set of real $m \times n$ matricies
$\mathbb{R}_{>0}$	=	Set of positive real numbers
$\mathbb{R}_{\geq 0}$	=	Set of nonnegative real numbers
$\mathbb{Z}$	=	Set of integers
$\mathcal{C}_{\mathrm{A}}$	=	Constraint set
$\mathcal{C}_{\mathrm{b}}$	=	Safe terminal set (backup set)

$\mathcal{C}_{\mathrm{S}}$	=	Safe set
$\inf$	=	Infimum
$\sup$	=	Supremum
$SO(3)$	=	3-dimensional rotation group
u	=	Control vector
$u_{\mathrm{act}}(x, u)$	=	RTA control law
$u_{\mathrm{b}}(x)$	=	Backup control law
$u_{\mathrm{des}}(x)$	=	Desired control law
U	=	Admissible control domain
w	=	Nondeterministic input variable
W	=	Nondeterministic input bounds
x	=	State
X	=	State domain

Acronyms and Initialisms

ASIF	=	Active Set Invariance Filter
CBF	=	Control Barrier Function
CPS	=	Cyber-Physical System
MIP	=	Mixed Integer Program
MPC	=	Model Predictive Control
NMT	=	Natural Motion Trajectory
OCP	=	Optimal Control Program
RTA	=	Run Time Assurance
QP	=	Quadratic Program

Summary

This dissertation investigates the problem of safety-critical control for complex cyber-physical systems, with an emphasis on numerical optimization and autonomy applications in the space domain. The first part of the work presents a tutorial on safety and Run Time Assurance (RTA). A set-based approach is presented for specifying mission constraints, and safety is formalized in the context of set invariance. Next, the research investigates the topic of RTA, which relates to a control system architecture where a performance-oriented controller is augmented with a safety-oriented element that filters the control signal in such a way that guarantees safety. The main contributions of this work are presented in the latter half of this thesis, and these relate to the application of optimization-based RTA techniques to problems in the space domain.

Autonomous rendezvous proximity operations and docking (ARPOD) is considered under proximity, collision-avoidance, and speed constraints. Natural motion trajectories are used to identify a set of passively safe *parking orbits* under the Clohessy-Wiltshire-Hill dynamics, and a mixed integer programming approach is used to generate safety-constrained optimal transfer trajectories to this set. The formulation is encoded into an RTA framework.

The safety problem is considered for a torque-controlled spacecraft in free rotational motion, subject to line-of-sight constraints. A nondeterministic dynamics model is considered and an RTA filter is constructed that relies on online computation of forward reachable sets around a recovery maneuver. The approach utilizes recent results from reachability theory in addition to optimization-based computation of invariant sets. Safety guarantees exist when a disturbance torque is bounded. The practicality of the approach is demonstrated with an application on a hardware testbed.

Finally, the research studies the topic of harnessing collisional behavior for free-flying spacecraft. A framework is proposed for collision-inclusive trajectory optimization. Experimental comparisons of trajectories with and without collision-avoidance requirements demonstrate the capability of the collision-inclusive strategy to achieve significant performance improvements in realistic scenarios. Additionally, a safety application is considered, and the planner is utilized for the purpose of optimally mitigating damage in the presence of an inevitable collision.

CHAPTER 1
INTRODUCTION

1.1 Safety-Critical Control

Intelligent cyber-physical systems (CPSs) offer an enormous amount of economic and scientific value. This value is at its core tied to the ability to automate complex and meaningful decisions. However, in order to make good decisions, it is first necessary to avoid bad ones. Likewise, in order to reap the benefits of intelligent decision making, it is necessary that the possibility of taking harmful actions be eliminated. Fundamentally, safety-critical control relates to the problem of avoidance in control and decision making. Results in this area require a high degree of formality and technical rigor. This rigor is achieved through mathematical abstraction, and it is necessitated by the fact that exceedingly rare events can negate the accumulated positive value of a technology that performs well nominally. Specifically, for the class of systems that are safety-critical, errors may be catastrophic; *e.g.* resulting environmental damage, significant financial loss, or loss of life.

The need for a strong safety-critical control framework is greater than ever. The emergence of new technologies for embedded systems has enabled increasingly high-level decisions to be performed by on-board software. As intelligent systems begin to interact with the world in more complex ways, they become increasingly vulnerable to error. This complexity often precludes the ability to prove correctness. Furthermore, as tasks traditionally performed by human operators continue to be automated, the burden of safety assurance shifts ever further toward the designer of the system. For mobile systems with high degrees of autonomy, safety poses the single greatest obstacle to widespread and practical utilization in the real world. In order to enable the seamless integration of autonomous robotics and other advanced mobile systems, it is necessary to develop methods for enforcing safety that are both rigorous enough to inspire the confidence required for real-world deployment, and scalable enough to the complexities of real-world systems.

1.2 Run Time Assurance

A controller design task generally consists of finding a control policy that maximizes some performance criteria while ensuring safety of the system. However, system performance and safety are often conflicting goals, and the need to solve both problems in tandem adds a great amount of complexity to the design process. A crucial observation is that the two problems can be decoupled. Run Time Assurance (RTA) presents an approach to control design whereby a performance-driven controller is augmented with a safety-driven element that preempts unsafe actions when necessary. Ideally, this element is minimally invasive to the desired input from the performance-driven controller, leaving the desired input unmodified whenever it is not compromising to system safety. This process allows for guarantees on safety to be made independently of the structure of the objectives of the primary performance-driven controller. Importantly, it provides a means for safety constraints to be enforced in systems being controlled by human operators. By separating safety and performance concerns, both problems are greatly simplified. A key benefit is that this approach allows for control laws for safety-critical CPSs to be complex, adaptive, and readily modified, without the need to repeat a rigorous verification process, and without compromising on safety. The research in this thesis follows an RTA paradigm, and as such, the results are complementary to those in performance-driven research.

1.3 Numerical Optimization

Numerical optimization provides an attractive framework for reliable control and decision making in safety-critical systems. Efficient off-the-shelf optimization interfaces allow users to take advantage of state-of-the art algorithms to solve constrained optimization problems specified at a high level. Furthermore, the algorithms can be formally verified [1, 2]. Optimization is utilized for safety-critical control in a number of ways. Model-Predictive Control (MPC) problems seek to synthesize safe inputs directly at the planning stage by optimizing a performance objective subject to a set of defined constraints on the state and input. Constraints may be added to model physi-

cal properties of the system (*e.g.* the equations of motion, actuator limitations), or they may be imposed by the designer to ensure that a safety specification is met (*e.g.* collision avoidance). Optimization may additionally be used in an RTA framework, where a verified safety filter is used to augment the potentially unsafe inputs of an unverified performance-driven controller in a way that ensures safety of the system when necessary. In this case, constraints restrict the input to the subset of available inputs that are safe. One may also consider the case where safety is reflected in the objective, such as when hard safety constraints are *softened*, meaning that constraint violations are allowed, but penalized in the objective function.

1.4 Mixed Integer Programming

Mixed Integer Programming (MIP) refers to an optimization problem that includes both real and integer-valued decision variables. This type of problem provides a very general framework for capturing many types of practical control objectives. Specifically, the inclusion of integer variables allows for both the expression of discrete decisions (*e.g.* modes or assignments) and the encoding of non-convex constraints (*e.g.* obstacle avoidance). This makes MIP naturally well suited to optimizing the actions over systems governed by interdependent dynamic modes, logical statements, and operational constraints [3, 4]. Though MIP may refer to any optimization problem involving both real and integer decision variables, the most commonly considered case entails an objective function $J(z)$ that is optimized over piece-wise affine (PWA) constraints, fitting the standard form below:

$$\begin{aligned} \min_{z} \quad & J(z) \\ \text{s.t.} \quad & D_{\mathrm{c}} z_{\mathrm{c}} + D_{\mathrm{b}} z_{\mathrm{b}} \leq g, \quad A_{\mathrm{c}} z_{\mathrm{c}} + A_{\mathrm{b}} z_{\mathrm{b}} = h \\ & z_{\mathrm{c}} \in \mathbb{R}^{n_{\mathrm{c}}}, \quad z_{\mathrm{b}} \in \{0,1\}^{n_{\mathrm{b}}}, \quad z = [z_{\mathrm{c}}, z_{\mathrm{b}}] \end{aligned} \tag{1.1}$$

where, $D_{\mathrm{c}} \in \mathbb{R}^{m \times n_{\mathrm{c}}}$, $D_{\mathrm{b}} \in \mathbb{R}^{m \times n_{\mathrm{b}}}$, $g \in \mathbb{R}^m$, $A_{\mathrm{c}} \in \mathbb{R}^{p \times n_{\mathrm{c}}}$, $A_{\mathrm{b}} \in \mathbb{R}^{p \times n_{\mathrm{b}}}$, $h \in \mathbb{R}^p$, and the constraint inequality symbols are element-wise. Note that the above constraints may be interpreted as a single *conjunction* of m inequality constraints and p equality constraints.

While the problem (1.1) is not convex in general, one can in principle compute globally optimal

solutions whenever J is convex by solving a finite number of convex subproblems [5]. This is apparent when one notes that the problem is convex for each fixed combination of assignments to the integer variables. For example, a brute force approach to solving (1.1) is to enumerate all 2^{n_b} integer values of z_b, and compare the optimal costs of each feasible subproblem [4]. By far the most common choice for solving MIP problems in practice is the *branch-and-bound* algorithm [6], which searches a *tree* with different integer settings for each branch. Branch-and-bound is known to return a globally optimal solution upon termination. While binary integer programming is in general $\mathcal{NP}$-complete [7, 8], and indeed the *worst-case* computation time of branch-and-bound is exponential in the number of binary variables[1], solutions to MIP problems can readily be found with good average-case performance using off-the-shelf optimization software; *e.g.* CPLEX [10], Gurobi [11], or MOSEK [12].

In addition to the ability to be solved to global optimality with certain objective functions, the attractiveness of MIP derives from the fact that mixed integer constraints can express a rich collection of practical specifications. Of central importance here is the procedure [3] for encoding propositional logical connectives as mixed integer linear constraints. That is, given a set of specifications consisting of mixed integer inequalities separated by logical connectives, there exists a separate set of mixed integer inequalities that is satisfied exactly when the first set of specifications is satisfied. For example, let $x \in \mathcal{X} \subset \mathbb{R}^n$, then the disjunction

$$a_1^T x \leq b_1 \quad \vee \quad a_2^T x \leq b_2 \tag{1.2}$$

is equivalent to the mixed integer linear constraints,

$$a_1^T x \leq b_1 + M z_1, \qquad a_2^T x \leq b_2 + M z_2, \qquad z_1 + z_2 \leq 1 \tag{1.3}$$

[1]When $J(z)$ is linear or quadratic, complexity is polynomial in the number of real variables [9]. Polynomial-time complexity also holds true for semi-definite programs (SDPs) and second-order cone programs (SOCPs). It is interesting to note that for the case of linear programs, the Simplex algorithm is more commonly used, which has an exponential worst case complexity but has better average case performance. See [1] for more on this topic.

where, $z_1, z_2 \in \{0, 1\}$, and $M \in \mathbb{R}_{>0}$ is defined to be sufficiently large such that the constraint is always satisfied when the integer variable that it is attached to is assigned a value of one; *e.g.* in the above case we set $M \geq \max_{x \in \mathcal{X}}(a_1^T x - b_1,\ a_2^T x - b_2)$. In practice, M should be chosen carefully, as excessively large values may decrease computational efficiency or introduce numerical error. The above procedure is commonly referred to in the literature as the *big-M* method. It provides a basis for representing more complex constraints such as obstacle avoidance. In addition, mixed integer linear constraints have been shown to equivalently represent specifications in Linear Temporal Logic (LTL) [13], and Signal Temporal Logic (STL) [9, 14]. As a result of this, the problem of synthesizing optimal control laws subject to high-level temporal logic specifications may be posed as a MIP.

1.5 Modelling Cyber-Physical Systems

Cyber-physical systems (CPSs) feature computing devices that interact with the physical world via actuators and sensors [15]. Mathematical models provide a means for interacting with CPSs through abstraction of real-world behavior, and these models may be defined at varying levels of complexity based on the contrasting needs of accurately predicting system behavior and conducting mathematical analysis. In this setting, CPSs are modeled as *dynamical systems*; hereafter the variable $x \in X \subseteq \mathbb{R}^n$ denotes the state vector of the system model and the variable $u \in U \subset \mathbb{R}^m$ denotes a bounded control input, where X is the set of all possible system states and the set U is the *admissible* set of controls, determined *a priori* by the actuation constraints of the real-world system.

Continuous-time system models appear as systems of ordinary differential equations as in

$$\dot{x} = F_{\mathrm{c}}(x, u), \tag{1.4}$$

and *discrete-time* system models appear as state-update maps as in

$$x^+ = F_{\mathrm{d}}(x, u). \tag{1.5}$$

While the computational elements responsible for the control of a CPS live in a discrete world, the physical laws of motion tend to take the form of ordinary differential equations. In many cases, systems of the form (1.5) are obtained from discretizing equations of the form (1.4) over the controller update period. A natural generalization is to consider *hybrid* systems that combine continuous-time and discrete-time elements [16, 17].

1.5.1 Nondeterministic Systems

It is important to observe that (1.4) and (1.5) are only models, and real-world systems generally deviate from their models. When this deviation is significant, safety guarantees derived from the models are invalid. One method for addressing this limitation is to instead consider a *nondeterministic model* that explicitly considers uncertainties that account for the deviation between the real-world system and the *deterministic models* presented in (1.4) or (1.5). This may be achieved by augmenting the above systems with a bounded nondeterminisitc input variable $w(t) \in W \subset \mathbb{R}^p$, which takes an *unknown* value within the domain W, and may be used to model parametric uncertainty, external noise perturbations, *etc.* Continuous-time nondeterministic models appear as in

$$\dot{x} = G_{\mathrm{c}}(x, u, w), \tag{1.6}$$

and *discrete-time* nondeterministic system models appear as in

$$x^+ = G_{\mathrm{d}}(x, u, w). \tag{1.7}$$

1.5.2 Admissible Control of Cyber-Physical Systems

Safety control schemes must conform to the boundaries of admissible control. That is, a control law can only be realized on a physical system if its outputs are bounded to a set of signals that respect the actuation constraints of that system. A feedback *control law* u denotes a mapping to $\mathbb{R}^m$ from either the state domain or an augmentation of the state domain. For instance, in the context of RTA, many primary control laws will take the form $u : X \to \mathbb{R}^m$, while RTA control laws will generally take the form $u : X \times \mathbb{R}^m \to \mathbb{R}^m$, and either representation can be extended to include explicit dependence on time. Control laws are said to be *admissible* when their range is the admissible input set U, *e.g.* $u : X \to U$ or $u : X \times \mathbb{R}^m \to U$.

Any control law u can be made admissible via composition with a saturation (also called clamping) function. A *saturation function* $\sigma : \mathbb{R}^m \to U$ is such that $\sigma(u)$ is approximately equal to u when u is in the interior of U and $\sigma(u)$ is approximately the projection of u onto the boundary of U when u is outside of U. There are two main approaches: (i) *hard clamping*, where the saturation function $\sigma(u) = u$ for all $u \in U$ and $\sigma(u)$ is otherwise a projection onto the boundary ∂U, and (ii) *smooth clamping*, whereby σ takes the form of a differentiable function that such that $\sigma(u) \approx u$ for $u \in U$.

Example 1.5.1. For the case of a scalar input and $U = [-1, 1] \subset \mathbb{R}$, then $\sigma_1(u) = \max(-1, \min(1, u))$ is a hard clamping function while $\sigma_2(u) = \tanh(\mathrm{u})$, $\sigma_3(u) = \frac{2}{\pi}\tan(\frac{\pi}{2}\mathrm{u})$, $\sigma_4(u) = u(1 + u^2)^{-1/2}$, and $\sigma_5(u) = \tanh(u + 0.5u^3)$ are examples of smooth clamping functions. ■

1.6 Safety and Invariance of Cyber-Physical Systems

In a colloquial context, safety means freedom from harm or danger. For safety critical systems, safety is freedom from conditions that would, in a worst-case environment, lead to an unacceptable loss, such as a loss of control, physical damage to the system under control, loss of human life, human injury, property damage, environmental pollution, or failure of the mission [18, 19]. In the context of a controlled dynamical system, safety is a characterization of the state of the system

and its evolution. While alternate approaches to forming safety specifications exist (*e.g.* temporal logic), this research focuses specifically on set invariance requirements derived from static properties on the state. Specifically, safety properties are specified with state constraints, and safety relates to whether a particular initial condition will lead to the safety constraints being satisfied for all time.

Safety constraints in this research are defined with inequality constraints on a function of the state. In particular, we consider $\varphi_i : \mathbb{R}^n \to \mathbb{R}$ for $i \in \{1, ..., M\}$, where M is the number of safety constraints and it is required that always $\varphi_i(x) \geq 0$ for each i. Though contextual, these constraints are usually made to reflect some form of harmful configuration, as described above. The set of states that satisfy all of the safety constraints is referred to as the *constraint set* —sometimes called the *allowable set* or the *constraint space*— and is given by

$$\mathcal{C}_{\mathrm{A}} := \{x \in \mathbb{R}^n \,|\, \varphi_i(x) \geq 0,\ i \in \{1, \cdots, M\}\}. \tag{1.8}$$

The construction of the constraint set is demonstrated in the following example.

Example 1.6.1. Consider a wingman aircraft with position $(x_{\mathrm{W}}, y_{\mathrm{W}})$ flying co-altitude in formation with a lead aircraft with position $(x_{\mathrm{L}}, y_{\mathrm{L}})$ and the requirement is that the two aircraft never go within 30 meters of each other. A constraint set over the states $x = [x_{\mathrm{L}}, y_{\mathrm{L}}, \dot{x}_{\mathrm{L}}, \dot{y}_{\mathrm{L}}, x_{\mathrm{W}}, y_{\mathrm{W}}, \dot{x}_{\mathrm{W}}, \dot{y}_{\mathrm{W}}]^T$ is given by:

$$\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R}^8 \,|\, \sqrt{(x_{\mathrm{L}} - x_{\mathrm{W}})^2 + (y_{\mathrm{L}} - y_{\mathrm{W}})^2} - 30 \geq 0\}. \tag{1.9}$$

■

Importantly, the identification of the constraint set is only half of the story, as there may exist states that obey the safety constraints at a given time, but that inevitably lead to violations in the future. For example, in the aircraft case it is possible for the wingman aircraft to have just more than 30 m separation with the lead aircraft initially, but be traveling too fast to to avoid a collision. Though a state may exist in the constraint set $\mathcal{C}_{\mathrm{A}}$ at a given time, it is not "safe" (in any meaningful sense of the word) if it inevitably leads to a departure from the set in the future. Hence, a definition

of safety must encode additional information relating to the whether the safety constraints will continue to be satisfied for all time with a particular control law, and subject to a particular set of dynamics and actuation constraints.

Informally, a system is safe when starting from a particular initial condition if the state is bounded to $\mathcal{C}_{\mathrm{A}}$ for all time. This concept is formalized with the notion of *set invariance*:

Definition 1.6.1 (forward invariance)**.** A closed set $\mathcal{S}$ is *forward invariant* with respect to a dynamical system if:

$$x(t_0) \in \mathcal{S} \implies \forall\, t \geq t_0, \quad x(t) \in \mathcal{S}. \tag{1.10}$$

△

That is, a set is forward invariant if any trajectory starting in that set will stay in that set for all time. Furthermore, trajectories may flow into the boundary of a forward invariant set, but not outwards. Using this definition, safety is defined as follows:

Definition 1.6.2 (safety)**.** A set of states $\mathcal{C}_{\mathrm{S}} \subset X$ is said to be *safe* with respect to a dynamical system, a control law, and a constraint set $\mathcal{C}_{\mathrm{A}}$, if that set is a forward invariant subset of the allowable set. That is, if $\mathcal{C}_{\mathrm{S}} \subseteq \mathcal{C}_{\mathrm{A}}$ and $x(t_0) \in \mathcal{C}_{\mathrm{S}} \implies \forall\, t \geq t_0,\ x(t) \in \mathcal{C}_{\mathrm{S}}$. Furthermore, if $\mathcal{C}_{\mathrm{S}}$ is a safe set, then any state in $\mathcal{C}_{\mathrm{S}}$ is said to be a safe state. △

In addition, control laws are said to be safe when they render some nonempty subset of the constraint space forward invariant. It is generally desirable that the safe set to be as large as possible. However, it is generally not possible to find an admissible control law that will render $\mathcal{C}_{\mathrm{A}}$ itself forward invariant.

It is important to note that forward invariance, and by extension safety, are properties of the closed-loop system and are not defined in absence of a controller. The existence, size, and shape of a safe set are all dependent on the controller. The question of whether it is *possible* to find a control law that will render a particular set invariant is addressed with the concept of control invariance.

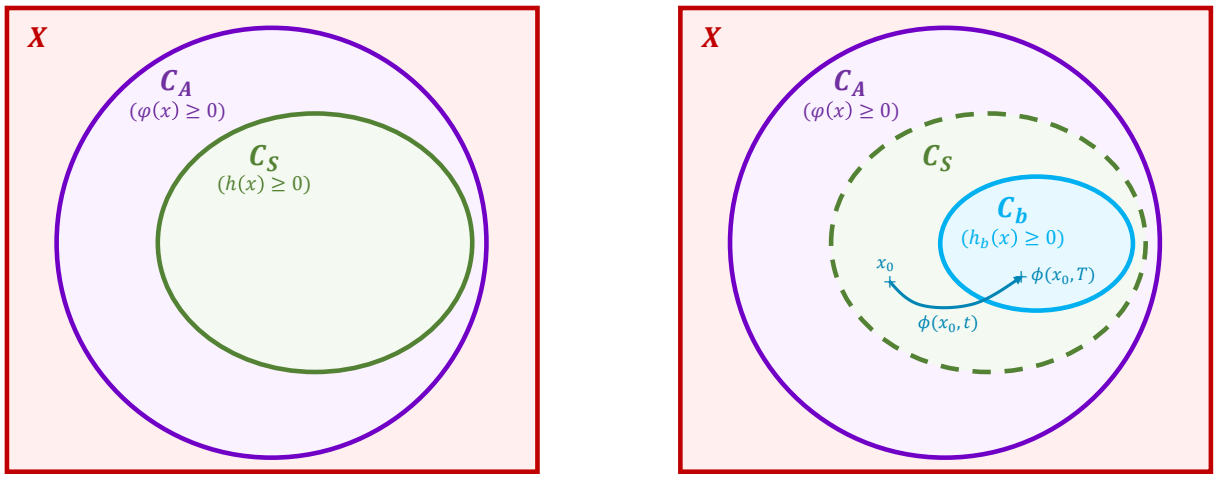


Figure 1.1: Safe set $\mathcal{C}_\mathrm{S}$ and constraint set $\mathcal{C}_\mathrm{A}$ topology for explicit (left) and implicit (right) cases.

Definition 1.6.3 (control invariance)**.** A closed set $\mathcal{S} \subseteq X$ is said to be *control invariant* (also called *viable*) with respect to a dynamical system and admissible control set U if there exists an admissible control law that renders it forward invariant. △

The largest control invariant subset of $\mathcal{C}_\mathrm{A}$ is known as the *viability kernel* [20]. Intuitively, the viability kernel is the largest achievable safe set, and it forms the boundary between the states for which it is and is not possible to find a control law that will keep the system safe for all time. A fact that will become important in later sections is that control invariant sets are forward invariant under controllers that take the form of certain optimization-based procedures.

Example 1.6.2. Consider the double integrator system —*e.g.* a spacecraft in linear motion— described by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{1.11}$$

where the state vector $x = [x_1, x_2]^T \in \mathbb{R}^2$ is composed of a position state x_1, and a velocity state x_2, and $u \in [-1, 1]$ is the acceleration. The safety constraint $\varphi(x) = -x_1 \geq 0$ is imposed on the

system, reflecting the requirement that the system avoid collision with an object located at $x_1 = 0$. The constraint set is

$$\mathcal{C}_{\mathrm{A}} := \{x \in \mathbb{R}^2 \mid -x_1 \geq 0\}. \tag{1.12}$$

The largest control invariant subset (*i.e.* the viability kernel) of $\mathcal{C}_{\mathrm{A}}$ is $\mathcal{C}_{\mathrm{S}} = \{x \in \mathbb{R}^2 \mid h(x) \geq 0\}$ where

$$h(x) = \begin{cases} -2x_1 - x_2^2 & \text{if} \quad x_2 > 0 \\ -x_1 & \text{if} \quad x_2 \leq 0. \end{cases} \tag{1.13}$$

The unsafe states in the constraint set $\mathcal{C}_{\mathrm{A}} \setminus \mathcal{C}_{\mathrm{S}}$ represent states for which a future collision is inevitable. Intuitively, $x_2 = \sqrt{-2x_1}$ represents the maximum safe velocity at x_1. If this approach speed is exceeded, then there will not be sufficient distance to stop before a collision occurs. This result is made apparent by considering the flow of the system under a recovery maneuver $u = -1$, which applies maximum control away from the obstacle. It can be seen that $\mathcal{C}_{\mathrm{S}}$ is a forward invariant set under this control law. Figure 1.2 shows a depiction of the described sets, and the flow under various inputs.

■

Implicit and Explicit Definitions of the Safe Set

Given a constraint set $\mathcal{C}_{\mathrm{A}}$ and an admissible control law $u : X \to U$, the question arises of how one can determine if a given state $x \in X$ is safe. In many cases, the set of safe states $\mathcal{C}_{\mathrm{S}}$ can be identified *explicitly* with a functional representation; *e.g.*

$$\mathcal{C}_{\mathrm{S}} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}, \tag{1.14}$$

with $h : \mathbb{R}^n \to \mathbb{R}$. In this case, checking whether $x \in \mathcal{C}_{\mathrm{S}}$ is equivalent to checking whether $h(x) \geq 0$. The safe set boundary may be determined in a number of ways, such as through Lyapunov arguments or reachability analysis. Additionally, for continuous-time systems as in (1.4), Nagumo's

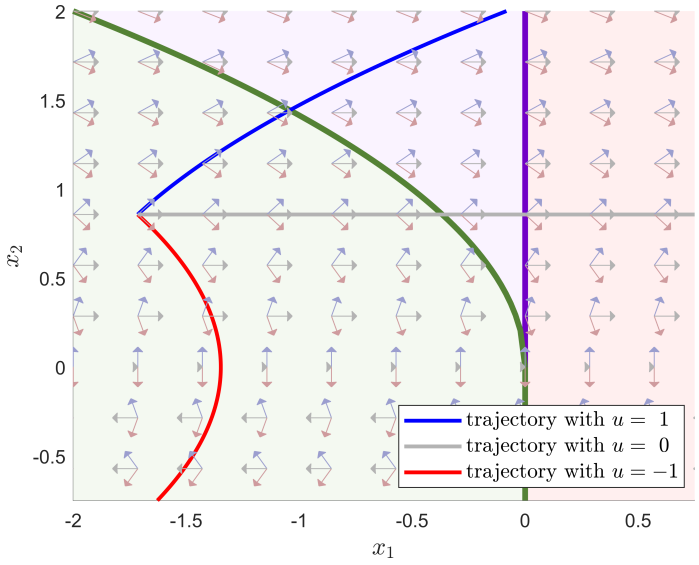


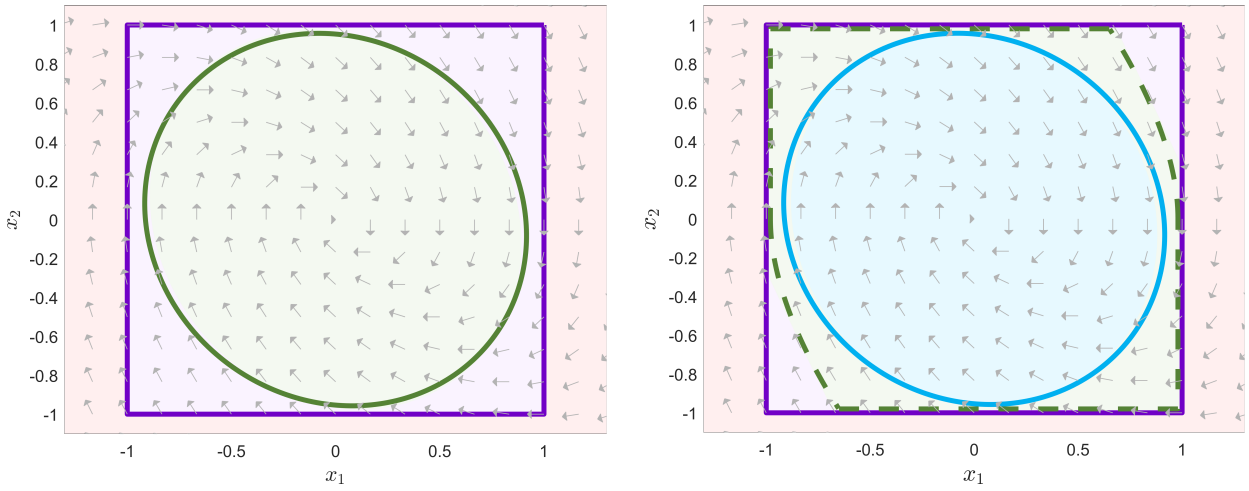
Figure 1.2: Phase plot for double integrator system with $u = -1, 0, 1$. The viability kernel is shaded green, collision states are shaded red, and the *inevitable* collision states are shaded purple.

theorem states that, under appropriate regularity assumptions on f, u_{b}, and h, a necessary and sufficient condition for invariance of a candidate set $\mathcal{C}_{\mathrm{S}}$ as in (1.14) is $\frac{\partial h}{\partial x}(x)^T F_c(x, u_b(x)) \geq 0$ for all x such that $h(x) = 0$ [21], and thus Nagumo's theorem can be used to verify that a candidate safe set is invariant. Moreover, it is possible to consider nondeterministic effects through the identification of sets that are *robustly* forward invariant; *i.e.* that are forward invariant under any realization of a nondeterminism.

Example 1.6.3. Consider the double integrator system (1.11), now paired with the constraint set $\mathcal{C}_{\mathrm{A}} := \{x \in \mathbb{R}^2 \,|\, 1 - \|x\|_\infty \geq 0\}$. Suppose that a control law is constructed such that for all $x \in \mathcal{C}_{\mathrm{A}}$, $u = -x_1 - x_2$. Note that actuation constraints can be ignored in $\mathcal{C}_{\mathrm{A}}$ if a hard clamping function is used, and that for all $x \in \mathcal{C}_{\mathrm{A}}$, $u(x) \in U$. In this case, the closed-loop dynamics are

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} x, \tag{1.15}$$

for all states in $\mathcal{C}_{\mathrm{A}}$. A valid safe set can be constructed by identifying a region of attraction con-



(a) Explicit safe set $\mathcal{C}_\mathrm{S}$ (green)

(b) Implicit safe set $\mathcal{C}_\mathrm{S}$ (green) and backup set $\mathcal{C}_\mathrm{b}$ (blue)

Figure 1.3: Phase plots for system (1.15) with the complement of the constraint space $\mathcal{C}_\mathrm{A}$ shaded red, and $\mathcal{C}_\mathrm{A} \backslash \mathcal{C}_\mathrm{S}$ shaded purple. The figure on the left corresponds to the sets described in Example 1.6.3, and the figure on the right corresponds to the sets described in Example 1.6.5.

tained in $\mathcal{C}_\mathrm{A}$. For instance, note that

$$V(x) = 1.2x_1^2 + 0.2x_1x_2 + 1.1x_2^2 \tag{1.16}$$

is a Lyapunov function for the closed-loop system and the level curve $V(x) = 1$ is contained in $\mathcal{C}_\mathrm{A}$. Thus, $\mathcal{C}_\mathrm{S} = \{x \in \mathbb{R}^2 \,|\, h(x) \geq 0\}$, where $h(x) = 1 - V(x)$ defines an invariant set contained in $\mathcal{C}_\mathrm{A}$. The sets $\mathcal{C}_\mathrm{S}$ and $\mathcal{C}_\mathrm{A}$ are depicted along with the system flow in Figure 1.3(a). Note that while $\mathcal{C}_\mathrm{S}$ is invariant, it is not the largest forward invariant set contained in $\mathcal{C}_\mathrm{A}$. It is said to be conservative. ■

Explicit identification of forward invariant subsets is typically obtained only at the expense of conservatism. In particular, the methods used to identify forward invariant sets are not generally scalable to complex and high dimensional systems, and the safe sets obtained with these methods may greatly underapproximate the largest safe set obtainable. However, an explicit (functional) representation of the safe set boundary is not always necessary. One may *implicitly* define $\mathcal{C}_\mathrm{S}$ in terms of the closed-loop trajectories under the control law. For example, consider a *backup* control law $u_\mathrm{b} : X \to U$, and let $\phi^{u_\mathrm{b}}(t; x)$ represent the state reached after starting at $x \in X$ and applying

u_{b} for t units of time. Then the set

$$\mathcal{C}_{\mathrm{S}} = \{x \in \mathbb{R}^n \,|\, \forall t \geq 0,\ \phi^{u_{\mathrm{b}}}(t;x) \in \mathcal{C}_{\mathrm{A}}\} \tag{1.17}$$

is by definition an invariant set under u_{b}, and it is entirely contained in $\mathcal{C}_{\mathrm{A}}$. The utility of this definition arises from the observation that it is possible to check whether *individual states* are safe simply by integrating the dynamics forward from those states. For example, if $\mathcal{C}_{\mathrm{A}} := \{x \in \ | \, \varphi(x) \geq 0\}$ then the following are equivalent:

$$(i)\ \forall t \geq 0,\ \phi^{u_{\mathrm{b}}}(t;x) \in \mathcal{C}_{\mathrm{A}}, \qquad (ii)\ \inf_{t\in[0,\infty)} \varphi(\phi^{u_{\mathrm{b}}}(t;x)) \geq 0 \tag{1.18}$$

It is interesting to note that in special cases where the infimum can be solved in closed form, the solution can be used to define an explicit safe set, as shown in the following example.

Example 1.6.4. Consider the system,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \cos(x_2) \\ u \end{bmatrix} \tag{1.19}$$

with $u \in [-1, 1]$, the constraint set $\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R}^2 \,|\, x_1 \geq 0\}$, and the admissible control law $u_{\mathrm{b}} = 1$. The solution for the flow from initial state $x_0 = [x_{0,1}, x_{0,2}]^T$ is given by,

$$\begin{bmatrix} \phi_1^{u_{\mathrm{b}}}(x_0,t) \\ \phi_2^{u_{\mathrm{b}}}(x_0,t) \end{bmatrix} = \begin{bmatrix} x_{0,1} - \sin(x_{0,2}) + \sin(x_{0,2}+t) \\ t + x_{0,2} \end{bmatrix} \tag{1.20}$$

and x_0 is safe if

$$\inf_{t\in[0,\infty)} \phi_1^{u_{\mathrm{b}}}(x_0,t) = x_{0,1} - \sin(x_{0,2}) - 1 \geq 0. \tag{1.21}$$

Using this information, a safe set under this maneuver can be defined explicitly as,

$$\mathcal{C}_{\mathrm{S}} = \{x \in \mathbb{R}^2 \,|\, x_1 - \sin(x_2) - 1 \geq 0\}. \tag{1.22}$$

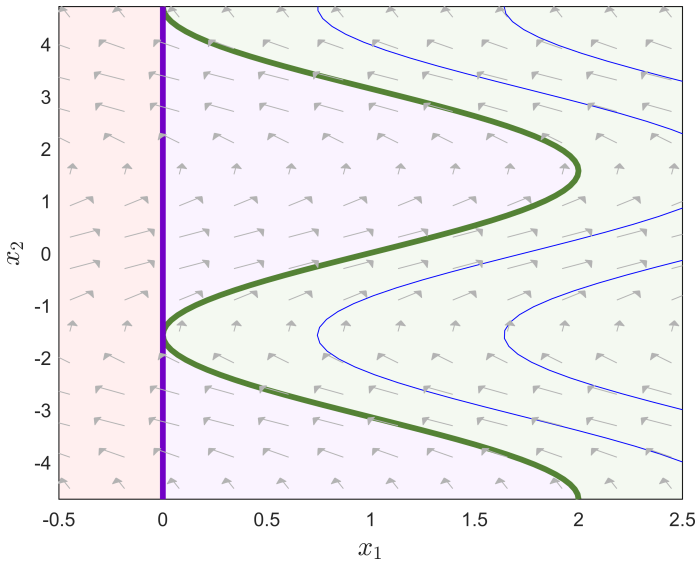


Figure 1.4: Phase plot for system (1.19) under $u_{\mathrm{b}} = 1$. The safe set $\mathcal{C}_{\mathrm{S}}$ is shaded green, the complement of $\mathcal{C}_{\mathrm{A}}$ is shaded red, and $\mathcal{C}_{\mathrm{A}} \backslash \mathcal{C}_{\mathrm{S}}$ is shaded purple.

The sets $\mathcal{C}_{\mathrm{A}}$, $\mathcal{C}_{\mathrm{S}}$ are depicted in Figure 1.4 along with the flow of the system under $u_{\mathrm{b}} = 1$. ■

Importantly, the system flow $\phi^{u_{\mathrm{b}}}(t; x)$ can readily be evaluated over a *finite* time horizon $\mathcal{T} = [0, T]$ by *numerically* integrating the dynamics. In this case, safety of a trajectory can be shown by ensuring that (i) the trajectory lies in $\mathcal{C}_{\mathrm{A}}$ over $\mathcal{T}$ and (ii) the endpoint of the finite trajectory T lies in a known invariant subset $\mathcal{C}_{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}}$. That is,

$$\mathcal{C}_{\mathrm{S}} = \{x \in \mathbb{R}^n \,|\, \forall t \in \mathcal{T},\ \phi^{u_{\mathrm{b}}}(t; x) \in \mathcal{C}_{\mathrm{A}} \wedge \phi^{u_{\mathrm{b}}}(T; x) \in \mathcal{C}_{\mathrm{b}}\}. \tag{1.23}$$

In this context, a safe terminal set $\mathcal{C}_{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}}$ is referred to as a *backup set*, and the set described by (1.23) is the *safe backward image* (SBI) of $\mathcal{C}_{\mathrm{b}}$. Note that numerical integration over a continuous interval $[0, T]$ requires that the trajectory be approximated with a finite amount of points sampled along this interval; *e.g.* $\mathcal{T} = \{t_1, ..., t_K\} \subset [0, T]$. However it is possible for the constraints defining $\mathcal{C}_{\mathrm{A}}$ can be tightened in such a way that accounts for this (see for example [22]). While a single case is presented here, various forms of implicit safety may arise under modified assumptions. For instance, [23] considers the case where $\mathcal{C}_{\mathrm{b}}$ is not forward invariant, and guarantees are in finite time. This concept may be extended to nondeterministic systems by approximating forward

reachable sets or invariant tubes around the recovery trajectories (see Chapter 4). The topology of the constraint set and safe set for both implicit and explicit cases is illustrated in Figure 1.1.

Example 1.6.5. Consider the system described by (1.15) along with the associated constraint set $\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R}^2 \,|\, \varphi(x) \geq 0\}$ with $\varphi(x) = 1 - \|x\|_\infty$. The safe set from the previous example is now taken as the backup set. That is, consider the backup set $\mathcal{C}_{\mathrm{b}} = \{x \in \mathbb{R}^2 \,|\, h_{\mathrm{b}}(x) \geq 0\}$ with $h_{\mathrm{b}}(x) = 1 - V(x)$ and $V(x)$ described by (1.16). The flow for this system can be evaluated exactly as

$$\phi^{u_{\mathrm{b}}}(t; x) = e^{At}x, \tag{1.24}$$

where A is the system matrix from (1.15). The backup time horizon $\mathcal{T} = [0, 3]$ is considered, and a state $x \in X$ is in the safe backward image if the following conditions are met,

$$(i)\ \forall t \in \mathcal{T}, \quad \varphi(e^{At}x) \geq 0, \qquad (ii)\ h_{\mathrm{b}}(e^{At_K}x) \geq 0. \tag{1.25}$$

The sets for this case are depicted in Figure 1.3(b). ■

Identifying Safe Backup Sets

Backup sets $\mathcal{C}_{\mathrm{b}}$ are safe terminal sets that act as *seeds* of safety. As seen in Figure 1.3 the backup set may be used to implicitly identify a larger safe region via trajectories under a backup controller. This is useful because small invariant sets are generally much easier to find than large ones. Furthermore, a finite-time trajectory can be used to show safety over an infinite horizon when it is shown that it the trajectory safely reaches such a set. This concept has been frequently explored in the context of trajectory optimization and *Model-Predictive Control* (MPC) [24, 25]. In the literature related to MPC, the backup set is often referred to as a *terminal feasible invariant set*. Backup sets *are* safe sets and may be computed as any safe subset of $\mathcal{C}_{\mathrm{A}}$ using the methods described above. However, it is common to construct these from domain knowledge of safe configurations of the particular system.

For an example of a safe backup set, consider that ground vehicles and vertical takeoff and

landing (VTOL) air vehicles may bring themselves to rest in a safe location. In this case, the role of u_{b} is to generate a safe stopping maneuver; *e.g.* [26]. Individual rest states are invariant points in the state space, and sets of adjacent rest states form invariant surfaces or volumes. In many cases it is not possible or practical to bring a vehicle to rest, and a backup set may instead be identified from safe periodic trajectories. For example, fixed wing aircraft may compute safe loiter circles [27, 28, 29, 25] or spacecraft may compute a set of safe orbits [30]. In many cases, the invariant set is a region without volume, and only an infinitesimal perturbation is required to cause a departure from the set. In practice, it is desirable to show that such surfaces are locally attractive under some backup control law, or to explicitly identify a larger invariant region around the initial set.

Example 1.6.6. Consider the damped linear system,

$$\ddot{x} = -\dot{x} + u \tag{1.26}$$

with the constraint set given by $\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R} \,|\, x \geq 0\}$. A safe backup set is given by the subset of $\mathcal{C}_{\mathrm{A}}$ with zero velocity: $\mathcal{C}_{\mathrm{b}} = \{x \in \mathbb{R} \,|\, x \geq 0, \dot{x} = 0\}$. Note that $\mathcal{C}_{\mathrm{b}} \subset \mathcal{C}_{\mathrm{A}}$ and that $\mathcal{C}_{\mathrm{b}}$ is invariant when $u = 0$. ■

Example 1.6.7. Consider two spacecraft in planar orbit around a central body: (i) a *target* spacecraft, which is in a fixed circular orbit with period τ, and (ii) a *chaser* spacecraft of mass m. In this setting, the dynamics of the chaser spacecraft are given by the Clohessy-Wiltshire-Hill equations, as developed in [31]:

$$\begin{aligned}
\dot{x}_1 &= x_3 \\
\dot{x}_2 &= x_4 \\
\dot{x}_3 &= 3n^2 x_1 + 2n x_4 + \tfrac{1}{m} u_1 \\
\dot{x}_4 &= -2n x_3 + \tfrac{1}{m} u_2 \\
\dot{x}_5 &= -|u_1| - |u_2|
\end{aligned} \tag{1.27}$$

with state $x \in \mathbb{R}^5$, and control input $u \in [-u_{\max}, u_{\max}]^2$. In this setting, $x_1,\ x_2$ denote the relative distances between the spacecraft, $x_3,\ x_4$ denote the relative velocities, and x_5 denotes a fuel state. Additionally, $n = \frac{2\pi}{\tau}$ [31]. The constraint set is defined by the constraints that the chaser must not run out of fuel, and must stay at a distance greater than $R_{\min}$ from the target spacecraft. The constraint set is $\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R}^5 \,|\, x_1^2 + x_2^2 - R_{\min}^2 \geq 0,\ x_5 \geq 0\}$.

Backup Set from Invariant Points: Invariant points exist as zero velocity states in the x_2-x_5-plane, hence $\{x \in \mathbb{R}^5 \,|\, x_1 = x_3 = x_4 = 0\}$ is invariant for $u \equiv 0$. A backup set is obtained from this set as $\mathcal{C}_{\mathrm{b},1} = \{x \in \mathbb{R}^5 \,|\, x_1, x_2, x_3 = 0, x_5 \geq 0, |x_2| \geq R_{\min}\}$.

Backup Set from Periodic Trajectories: It can be shown that the chaser spacecraft is on an elliptical orbit around the target spacecraft whenever $u = 0$ and $x_3 = \frac{n}{2}x_2,\ x_4 = -2nx_1$, with each individual orbit occupying the position states $x_1^2 + \frac{1}{4}x_2^2 = b^2$ where $b \geq 0$ is the semi-minor axis of the orbit. Hence, the linear subspace $\{x \in \mathbb{R}^5 \,|\, x_3 = \frac{n}{2}x_2,\ x_4 = -2nx_1\}$ is an invariant manifold composed of all such closed orbits. A backup set can be constructed from this subspace by considering range of orbits that fall into the constraint set: $\mathcal{C}_{\mathrm{b},2} = \{x \in \mathbb{R}^5 \,|\, x_3 = \frac{n}{2}x_2,\ x_4 = -2nx_1,\ x_1^2 + \frac{1}{4}x_2^2 \geq R_{\min}^2\}$. Chapter 3 considers a similar problem to this in more detail. ■

CHAPTER 2

RUN TIME ASSURANCE

RTA Systems are online verification mechanisms that filter an unverified *primary* controller output to ensure system safety. The primary control may come from a human operator, an advanced control approach, or an autonomous control approach that cannot be verified to the same level as simpler control designs. The critical feature of RTA systems is their ability to alter unsafe control inputs explicitly to assure safety. An important quality of an RTA system is that the assurance mechanism is constructed in a way that is entirely agnostic to the underlying structure of the primary controller. By effectively decoupling the enforcement of safety constraints from performance-related objectives, RTA offers a number of useful advantages over traditional (offline) verification.

A significant benefit of RTA is that it eliminates the need to design the primary controller in a way that conforms to traditional safety standards, which may not be directly applicable to the primary control design. Practical consequences of this are that RTA provides a means of testing new control algorithms on hardware platforms without compromising safety, and that it provides a potential *near-term certification path* for autonomous controllers and safety-critical human-controlled systems. Additionally, the verification of an assurance mechanism is generally simpler than verification of a performance-based controller, as it does not require consideration of the potentially complex performance-related objectives. For example, safety verification does not become more difficult as the primary controller grows more complex, and the verification process does not need to be repeated when changes are made to the primary controller.

The rise in popularity of RTA can be attributed to many factors, including: (i) the emergence of mobile autonomous systems operating in safety critical environments, *e.g.* away from very specialized and removed applications to the vicinity of humans; (ii) the increasing complexity of the primary control system in these systems making traditional verification techniques prohibitively

difficult; (iii) the growing desire to quickly update complex system software without compromising safety or repeating a costly verification process; (iv) the ability to perform fast computation online, *e.g.*, integrating trajectories or computing reachable sets; (v) the emergence of control barrier function (CBF) and active set invariance filtering (ASIF) methods that give smooth and *minimally invasive* modifications to the desired actions. Additionally, RTA seems to provide an attractive solution to bounding the behavior of machine learning and artificial intelligence (AI) systems. For example, while some notable efforts have been made in the direction of generating and verifying provably safe deep neural network controllers [32, 33, 34, 35], the complexity and dynamic nature of these systems often precludes the ability to generate rigorous safety guarantees on the primary controller. The RTA paradigm enables one to forgo this route entirely, without a need to compromise on safety.

2.1 Run Time Assurance for Safety-Critical Cyber-Physical Systems

Cyber-physical systems are said to be *safety-critical* when failure would result in loss of life, damage to property, or other unacceptable damages such as environmental harm [36]. An RTA architecture acts as an online-verification and enforcement tool for cyber-physical systems, and guarantees certain system-level safety requirements are met at run time.

2.1.1 The Run Time Assurance Architecture

RTA presents an approach to control design that allows a designer to sidestep the common trade-off between performance and safety. The central idea behind RTA is to decouple the task of enforcing safety constraints from all other objectives of the controller. This is achieved by splitting the control system into two components: a performance-driven *primary* controller, and a safety-driven *RTA mechanism*. The RTA mechanism preempts the desired inputs from the primary controller when necessary for ensuring the safety of the system, and otherwise lets the input pass through unaltered. This approach is associated with the feedback control architecture shown in Figure 2.1(b). The output of the performance-driven controller is referred to as the *desired* input and assigned the

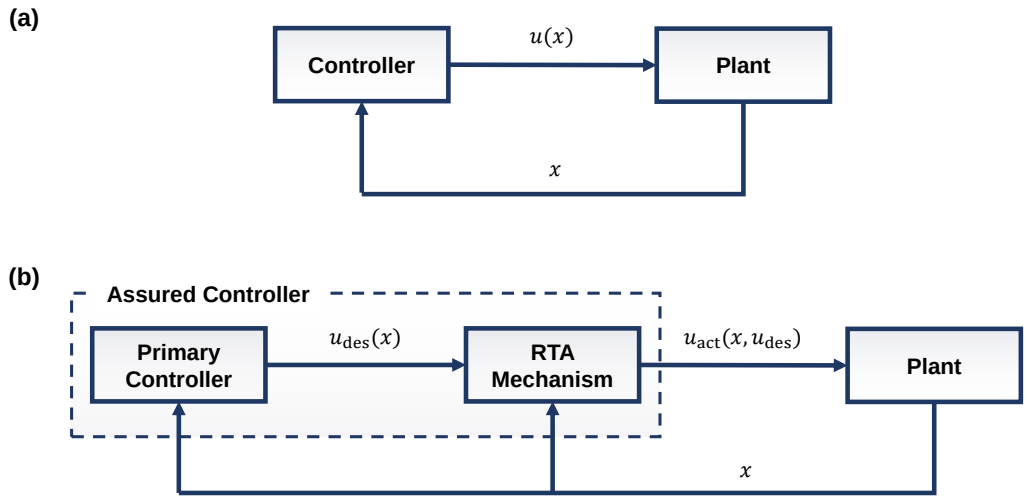


Figure 2.1: (a) Prototypical feedback control system architecture (b) Run time assurance architecture.

variable u_{des}, and the output of the RTA mechanism is referred to as the *actual* input (or assured input) and is assigned the variable u_{act}. Fundamentally, an RTA mechanism is a mapping from a state $x \in \mathbb{R}^n$ and a desired input $u_{\text{des}} \in U$ to an actual input $u_{\text{act}} \in U$, and has the objectives of (i) enforcing safety of the system as defined by state constraints, and (ii) having u_{act} as *close* as possible to u_{des}, so long as it does not conflict with the first objective. A common choice for measuring how *close* the two signals are is the norm of the difference between the two variables. However, this measurement may also take other forms, such as the integral of the deviation. An RTA mechanism is said to be *assured* when the output u_{act} renders a particular set of states safe. Moreover, an assured RTA mechanism guarantees safety for the entire system, and for all time, and this is true regardless of the chosen primary controller.

2.2 Properties of Run Time Assurance Systems

2.2.1 Implicit and Explicit RTA Approaches

An RTA mechanism is a control law that renders a subset of the constraint space forward invariant, that activates a recovery response near the boundary of that set, and that passes through the desired

input from the primary controller everywhere else. The set made safe under the RTA is referred to as the *safe operational region* of the RTA system and it is denoted by $\mathcal{C}_{\mathrm{S}}$. More precisely:

Definition 2.2.1. (safe operational region) Given a dynamical system, a constraint set $\mathcal{C}_{\mathrm{A}}$, and an RTA control law $u_{\mathrm{act}} : X \times U \to U$, the *safe operational region* of the RTA system refers to the largest subset of $\mathcal{C}_{\mathrm{A}}$ that is forward invariant under u_{act}. $\triangle$

A canonical approach to constructing an RTA mechanism is to find a safe *backup* control law, and to require that this control law be active near the boundary of the set that is forward invariant under this control law. In this case, the safe operational region is a forward invariant set under u_{b}. Approaches of this type are considered in Section 2.3. Alternatively, in cases where the recovery response is determined as the solution to an optimization program, and safety is enforced with the constraints of that program, then the safe operational region is a control invariant subset of $\mathcal{C}_{\mathrm{A}}$. This is due to the fact that the optimization program searches over all feasible control actions —*i.e.* control invariant sets are forward invariant under certain optimization procedures. Approaches of this nature are considered in Section 2.4.

As with the safe sets discussed previously, the safe operational region can be identified explicitly or implicitly. *Implicit* (or trajectory-based) approaches compute finite-time trajectories $\{\phi^{u_{\mathrm{b}}}(t;x) \in X \mid t \leq \mathcal{T}\}$ under a backup controller *online* (at run time) and use the information in the decision of when or how to intervene. *Explicit* (or region-based) approaches may or may not utilize a backup controller, but do not rely on simulating trajectories of the backup controller online. Typically, explicit approaches identify trusted regions *offline* via the construction of either a large forward invariant or control invariant set. That is, implicit RTA approaches are those that utilize an online *look-ahead* and will bound the system to an implicitly defined safe set such as the one given in (1.23) while explicit RTA approaches determine safe states *a priori* and will bound the system to an explicitly defined safe set such as the set described by (1.14). This concept is demonstrated in the following example.

Example 2.2.1. Consider the system described by (1.15) with the constraint space $\mathcal{C}_{\mathrm{A}} := \{x \in \mathbb{R}^2 \mid 1 - \|x\|_\infty \geq 0\}$. An RTA mechanism can be constructed by requiring that the backup controller

be applied near the boundary of a set $\mathcal{C}_{\mathrm{S}}$, where $\mathcal{C}_{\mathrm{S}}$ is safe under the backup controller. Let the RTA mechanism be,

$$u(x) = \begin{cases} u_{\mathrm{b}} & \text{if} \quad \vartheta(x) \\ u_{\mathrm{des}} & \text{otherwise,} \end{cases} \tag{2.1}$$

where u_{des} is the desired control input and the predicate $\vartheta : X \to \{\mathrm{true}, \mathrm{false}\}$ represents the *activation condition* (or switching condition). Considered below are implicit and explicit conditions.

Explicit Condition

Let $\mathcal{C}_{\mathrm{S}} = \{x \in X \mid h(x) \geq 0\}$ with $h(x) = 1 - V(x)$ and $V(x)$ being the Lyapunov function (1.16). The following activation condition ensures safety of $\mathcal{C}_{\mathrm{S}}$ with respect to u_{act} by activating u_{b} near the boundary,

$$\vartheta(x) : h(x) \geq \varepsilon \tag{2.2}$$

where $\varepsilon > 0$ defines the size of the RTA activation region. Since, $V(x)$ defines a Lyapunov function for the closed-loop dynamics under u_{b}, it is known that $h(x)$ is increasing whenever u_{b} is applied. Furthermore, since u_{b} is activated before the boundary, (2.1) enforces positivity of $h(x)$, meaning that the system will be forward invariant in $\mathcal{C}_{\mathrm{S}} \subset \mathcal{C}_{\mathrm{A}}$.

Implicit Condition

Let $\mathcal{C}_{\mathrm{b}} = \{x \in X \mid h(x) \geq 0\}$ with $h(x) = 1 - V(x)$ and $V(x)$ being the Lyapunov function (1.16), then a activation condition for invariance in the safe backward image of $\mathcal{C}_{\mathrm{b}}$ is

$$\vartheta(x) : \ [\forall t \in [0, T], \quad \varphi(e^{At}x) \geq \varepsilon_1] \ \wedge \ [h_{\mathrm{b}}(e^{At_K}x) \geq \varepsilon_2] \tag{2.3}$$

where $\varepsilon_1, \varepsilon_2 > 0$. The safe backward image in this case defines the safe operational region for the RTA. It is depicted as the green region in Figure 1.3(b). ■

There are important tradeoffs associated with implicit and explicit approaches. While identifying invariant sets explicitly offline generally reduces the online computational burden of the

algorithm, the task may become intractable for complex and high dimensional systems. In such cases, implicit (trajectory-based) approaches become favorable. The key advantage to assessing safety through trajectories online is that, in the simplest case, it only requires a finite-time simulation of the recovery maneuver, which is generally a tractable task. A secondary benefit is that it handles dynamic environments and changes in the model better; *i.e.* rather than needing to recompute an invariant subset of $\mathcal{C}_{\mathrm{A}}$, one only needs to update the simulation parameters. However, an important practical consideration is that implicit approaches rely on the ability to accurately forecast recovery trajectories, and the predicted behavior may be increasingly sensitive to noise (*i.e.* less accurate) as the length of the backup trajectory is increased. One way to address this is to simulate the reachable set around a recovery trajectory. See Chapter 4 for an example approach.

It is interesting to note that since a backup set $\mathcal{C}_{\mathrm{b}}$ is an explicitly defined invariant subset of the constraint set $\mathcal{C}_{\mathrm{A}}$, an implicit approach to safety will typically involve a combination of (i) explicitly identifying an invariant set $\mathcal{C}_{\mathrm{b}}$ offline (ii) finding a safe trajectory to this set online. In this sense, the explicit approach can be viewed as a special case of the implicit approach where the backup trajectory consists of only a single point. Furthermore, any explicitly defined safe set can be used as part of an implicit approach. The implicit approach allows for the problem to be solved in a way that takes advantage of both offline and online resources. Specifically, finding a larger backup set offline reduces the online computational burden by reducing the length of the trajectory required to obtain the same operational region; likewise, integrating backup trajectories over a longer horizon reduces the need to identify large safe sets offline. Generally, for simple systems, the problem of identifying invariant sets can be solved offline. As systems become more complex, one must rely more and more on online simulations.

2.2.2 Zero and First Order Methods

Filtering approaches in RTA may be classified into zero-order and first-order methods. Zero-order algorithms are derivative free. They are typically, though not necessarily, associated with the *Simplex* architecture and choosing from a discrete set of possible control signals. For example,

the RTA mechanism may choose between applying the desired signal or backup signal at any time. In this case, one may observe the effects of chatter; *i.e.* variations in the output of the RTA mechanism are not smooth with respect to variations in the initial state or desired input. This may have practical implications. For instance, the action may cause excessive wear to the actuators, on-board components may experience interruption to their operations, or passengers may observe the intervention as being more abrupt, resulting in a less pleasant experience (*e.g.* consider an automobile that only intervenes through applying the brakes at maximum capacity). These effects may be handled with various heuristics; *e.g.* hysteresis, or interpolating between u_{b} and u_{des} as the state approaches the boundary of the safe operational region. Alternatively, one may turn to a first order method.

First order methods require derivatives, and are typically associated with methods described Section 2.4. In this case, first-order information is used in a *barrier condition*, which constrains the set of available inputs to a subset of admissible and safe inputs. An optimization framework may be used to choose the input within this set according to some cost function. A common cost specification requires that the RTA choose the control that is closest (in a 2-norm sense) to the desired control. RTA filters constructed in this way will have smooth variations in the output. While this eliminates the effect of chatter, it introduces added complexity into the problem and generally requires greater computational resources.

Table 2.1: Classification of RTA applications in the literature.

	Implicit	Explicit
Simplex	spacecraft collision avoidance [37] fixed-wing aircraft ground avoidance (Auto-GCAS) [39, 40] fixed-wing aircraft obstacle avoidance [25, 28, 42] safety against LTL specifications [44, 45] collision avoidance for high speed quadrotor navigation [49, 26] collision avoidance during lane-changing maneuvers [52]	waypoint tracking cyber-physical system [38] advanced flight control systems [41] lane keeping [43] aircraft conflict resolution [46, 47, 48] decentralized collision avoidance for quadrotors [50, 51] ground robot navigation around obstacles [53, 54]
ASIF	bipedal walking/ exoskeletons [55] robotic manipulator arms [58] mobile inverted pendulum (Segway) [23] rapid aerial exploration of unknown environments [62] multi-robot systems [64] fixed-wing aircraft collision avoidance [69]	bipedal walking/ exoskeletons [56, 57] lane keeping [59, 60] mobile inverted pendulum (Segway) [61] robotic grasping [63] swarm robotics (Robotarium) [65, 66, 67, 68] motorized rehabilitative cycling system [70]

2.3 The Simplex Architecture

The Simplex architecture [71, 72, 38, 73, 44] relates to an RTA design where safety is guaranteed by using decision logic to activate a backup controller. That is, a Simplex RTA will at any time, apply either the desired input u_{des} or the input from a backup controller u_{b}. Without loss in generality, multiple backup controllers may be considered. One way to describe Simplex RTAs is as a hybrid dynamical system, such as (2.1). Though it is common in practice to verify the backup controller, this is not required in order to verify the RTA system as a whole. A depiction of the Simplex architecture is shown in Figure 2.2.

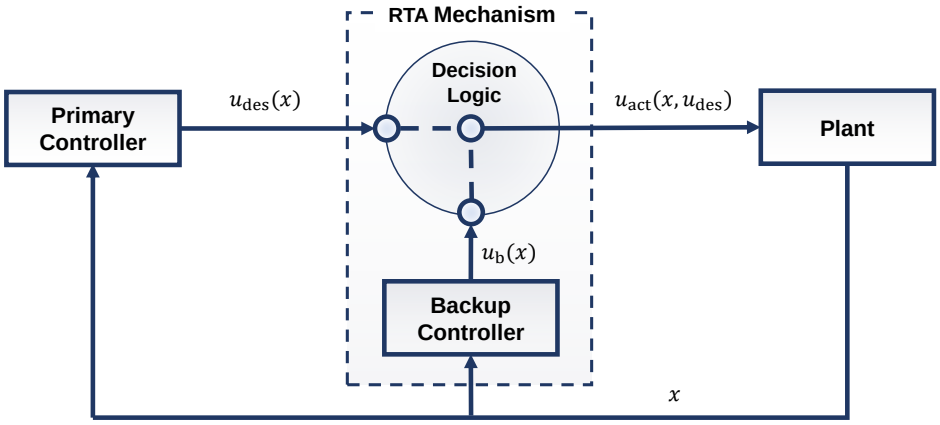


Figure 2.2: Simplex architecture with a physical plant, a backup controller, decision logic, and a primary controller.

2.3.1 Latched and Unlatched Implementations

Simplex RTA may be latched or unlatched. In an *unlatched* implementation, the decision logic releases control of the recovery system as soon as the activation condition becomes inactive. For example, (2.1) describes an unlatched RTA implementation that switches from the primary control law u_{des} to the backup control law u_{b} whenever a condition ϑ is active. Alternatively, a *latched* implementation will hold its activation state until a separate release condition is met. The release condition may for example consist of following the backup trajectory for some fixed amount of

time, bringing the system to a particular configuration, or having a human operator return control after assessing the problem. Other implementations of latched RTAs may require a system reset before switching back, such as a computer reset supervised by a human operator.

2.3.2 Provably Safe RTA with Black Box Backup Controllers

Implicit RTA systems rely on forecasting recovery trajectories under a backup controller in order to assess safety. It is important to note that the backup controller does not need to be verified in order to guarantee safety of the RTA system. This concept can be made apparent when one notes that the system can regulate to previously computed safe trajectories in the event that the backup controller fails to generate a safe solution. Under this paradigm, the task of the backup controller is to *propose* safe trajectories which the monitor then verifies online. A practical consequence of this is that high-performing black-box controllers can safely be used to generate backup trajectories. This concept is well known in practice [27] and is thoroughly explored in the context of Simplex-based systems in [74]. Furthermore, the idea extends to other methods of RTA. For example, in Section 2.4.2, an ASIF is constructed which utilizes a backup controller taking the form of a neural network approximation of an optimal control policy.

2.3.3 Canonical Algorithms

Included in this section are a set of canonical algorithms for Simplex-based RTA. In this case, we consider the deterministic discrete time system,

$$x(i+1) = F(x(i), u(i)) \tag{2.4}$$

where $x \in X \subseteq \mathbb{R}^n$ denotes the state, $u \in U \subset \mathbb{R}^m$ denotes the control input and $i \in \mathbb{Z}$ denotes the time index. A constraint set is defined as $\mathcal{C}_{\mathrm{A}} \subseteq X$. It is assumed that a backup control law $u_{\mathrm{b}} : \mathbb{R}^n \to U$ is defined and we let $\phi^{u_{\mathrm{b}}}(i; x)$ be the state of (2.4) reached at $i \geq 0$ when beginning at state x at $i = 1$ and evolving under u_{b}.

The idea behind the algorithms in this section is to assess the safety of a *probe step*. That is, at any state $x_{\mathrm{curr}} \in X$, whether to accept or reject the desired control input $u_{\mathrm{des}} \in U$ is based on whether the candidate next state reached under this input $x_{\mathrm{cand}} := F(x_{\mathrm{curr}}, u_{\mathrm{des}})$ is safe. In the first case, an explicit safe set is assumed to be defined. The second case considers an implicit approach, which relies on online simulations under the backup dynamics. A deterministic discrete time system is assumed for clarity of the algorithms and their associated guarantees. However, the algorithms may be adapted to continuous time systems, to multiple backup controllers, latched implementations, etc.

Explicit Simplex Filter

Algorithm 1 describes an explicit Simplex-based RTA algorithm, the Region-Based Simplex Filter (RBSF). In addition to a backup control law u_{b}, it is assumed that a safe set $\mathcal{C}_{\mathrm{S}} \subseteq \mathcal{C}_{\mathrm{A}}$ is defined such that $\mathcal{C}_{\mathrm{S}}$ is invariant under u_{b}. The algorithm renders the RTA system forward invariant in $\mathcal{C}_{\mathrm{S}}$. This is apparent through the observations that (i) by nature of the invariance property, applying u_{b} anywhere in $\mathcal{C}_{\mathrm{S}}$ will lead to a next state that is still in the set (ii) the sample step allows for inputs that would cause a departure from $\mathcal{C}_{\mathrm{S}}$ to be detected and replaced with u_{b}. In practice, one may choose to add conservatism to the approach by working with any underappoximation $\tilde{\mathcal{C}_{\mathrm{S}}} \subseteq \mathcal{C}_{\mathrm{S}}$. The region $\mathcal{C}_{\mathrm{S}} \setminus \tilde{\mathcal{C}_{\mathrm{S}}}$ is often referred to as a *buffer*. A practical consideration is that the backup controller should locally attract solutions outside of $\tilde{\mathcal{C}_{\mathrm{S}}}$ to $\tilde{\mathcal{C}_{\mathrm{S}}}$ near the boundary. That is, it is favorable to ensure that a recovery is possible in the case where the state is perturbed outside of the safe region. One way to do this is to ensure that all initial conditions in the buffer will lead to $\tilde{\mathcal{C}_{\mathrm{S}}}$ if u_{b} is applied; *e.g.* this is the case when $\mathcal{C}_{\mathrm{S}}$ and $\tilde{\mathcal{C}_{\mathrm{S}}}$ are both sublevel sets of a Lyapunov function.

Example 2.3.1. Consider the double integrator system described by (1.11) discretized over a period of 0.1 s, and consider the constraint set (1.12). The backup control law $u_{\mathrm{des}} = -1$ renders the set $\mathcal{C}_{\mathrm{S}} = \{x \in \mathbb{R}^2 \mid -2x_1 - x_2^2 \geq 0\}$ forward invariant and it is clear that $\mathcal{C}_{\mathrm{S}} \subseteq \mathcal{C}_{\mathrm{A}}$. Figure 2.7(a) shows a simulation of the RBSF algorithm with these parameters under the desired control $u_{\mathrm{des}} = 1$. ■

Algorithm 1: Region-Based Simplex Filter (Explicit Simplex)

input : Current State $x_{\text{curr}} \in X$
: Desired Input $u_{\text{des}} \in U$
output : Safe Control Input $u_{\text{act}} \in U$
predefined: Constraint set $\mathcal{C}_{\text{A}} \subset X$
: Invariant Set $\mathcal{C}_{\text{S}} \subset \mathcal{C}_{\text{A}}$
: Backup Control Law $u_{\text{b}} : \mathbb{R}^n \to U$.

1: **function** $u_{\text{act}} = \text{RBSF}(x_{\text{curr}}, u_{\text{des}})$
2: $x_{\text{cand}} \leftarrow F(x_{\text{curr}}, u_{\text{des}})$
3: **if** $x_{\text{cand}} \in \mathcal{C}_{\text{S}}$ **then**
4: **return** u_{des}
5: **else**
6: **return** u_{b}

Implicit Simplex Filter

Algorithm 2 describes an implicit Simplex-based algorithm, the Simulation-Based Simplex Filter (SBSF). It is assumed that a backup set $\mathcal{C}_{\text{b}} \subseteq \mathcal{C}_{\text{A}}$ is defined, and required that $\mathcal{C}_{\text{b}}$ is invariant under a backup control law $u_{\text{b}} : X \to U$. The controller u_{b} may come from a closed-form control expression, a black box function, an MPC, a motion primitive, *etc*. In contrast to the RBSF algorithm, which constrains the system to evolve in an explicitly defined invariant region, SBSF uses the smaller invariant set $\mathcal{C}_{\text{b}}$ as a means to access a larger implicitly defined invariant set. Specifically, SBSF constrains the system to the safe backward image of $\mathcal{C}_{\text{b}}$:

$$\mathcal{C}_{\text{S}} = \{x \in X \,|\, \phi^{u_{\text{b}}}(i; x) \in \mathcal{C}_{\text{A}} \ \forall i \in \{1, ..., N-1\},\ \phi^{u_{\text{b}}}(N; x) \in \mathcal{C}_{\text{b}}\}, \tag{2.5}$$

where, $\phi^{u_{\text{b}}}(i; x)$ is the i^{th} point in an N point trajectory obtained from simulating the dynamics (2.4) under a backup controller u_{b} from an initial state x. Since $\mathcal{C}_{\text{S}}$ is forward invariant under u_{b}, the algorithm renders the RTA system forward invariant in $\mathcal{C}_{\text{S}}$. As in the case of the RBSF algorithm, this becomes apparent from the observations that (i) applying u_{b} anywhere in $\mathcal{C}_{\text{S}}$ will lead to a state that is also in $\mathcal{C}_{\text{S}}$ and (ii) the sample step under u_{des} can reliably be used to detect inputs that would cause a departure from $\mathcal{C}_{\text{S}}$. Conservatism can be added to this approach by using

Algorithm 2: Simulation-Based Simplex Filter (Implicit Simplex)

input : Current State $x_{\text{curr}} \in X$
: Desired Input $u_{\text{des}} \in U$
output : Safe Control Input $u_{\text{act}} \in U$
predefined: Constraint Set $\mathcal{C}_{\text{A}} \subset X$
: Invariant Backup Set $\mathcal{C}_{\text{b}} \subset \mathcal{C}_{\text{A}}$
: Backup Control Law $u_{\text{b}} : \mathbb{R}^n \to U$.

function u_{act} =SBSF(x_{curr}, u_{des})
  $x_{\text{cand}} \leftarrow F(x_{\text{curr}}, u_{\text{des}})$
  compute: $\phi^{u_{\text{b}}}(i; x_{\text{cand}}) \quad \forall i \in \{1, \dots, N\}$
  if $\phi^{u_{\text{b}}}(i; x_{\text{cand}}) \in \mathcal{C}_{\text{A}} \quad \forall i \in \{1, \dots, N-1\}$ AND $\phi^{u_{\text{b}}}(N; x_{\text{cand}}) \in \mathcal{C}_{\text{b}}$
    return u_{des}
  else
    return u_{b}

under-approximations of $\mathcal{C}_{\text{A}}$ and $\mathcal{C}_{\text{b}}$. Note that the above algorithm assumes a deterministic system and relies on simulating single trajectories under the backup dynamics. This can be extended to the nondeterministic case by simulating an overapproximation of the forward reachable set under the backup dynamics, and requiring that $\mathcal{C}_{\text{b}}$ is robustly forward invariant. A nondeterministic formulation is considered in Chapter 4.

Example 2.3.2. Consider the double integrator system described by (1.11) discretized over a period of 0.1 s, and consider the constraint set (1.12). The backup control law $u_{\text{des}} = -1$ renders the set $\mathcal{C}_{\text{b}} = \{x \in \mathbb{R}^2 \mid -x_1 \geq 0, -x_2 \geq 0\}$ forward invariant and it is clear that $\mathcal{C}_{\text{b}} \subseteq \mathcal{C}_{\text{A}}$. Figure 2.7(b) shows a simulation of the SBSF algorithm with these parameters under the desired control $u_{\text{des}} = 1$. ■

2.4 Active Set Invariance Filtering

Active set invariance filtering (ASIF) methods are associated with a class of first-order, optimization-based RTA algorithms. As with many Simplex-based techniques, they are system agnostic, provably correct, and exhibit a number of robustness properties which make them attractive in practice. In addition, ASIFs are minimally invasive with respect to the safety constraints. This results in

a smoother and more gradual intervention than with approaches that rely on switching to backup controllers directly.

This section considers systems of the form

$$\dot{x} = f(x) + g(x)u. \tag{2.6}$$

When (1.4) is of the form (2.6), the dynamics are said to be *control affine*, and it is assumed throughout the following that $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipshitz continuous in their inputs so that, in particular, solutions to (2.6) are unique when they exist. ASIFs are constructed as quadratic programs. A common choice for the objective function is the l^2 norm of the difference between the desired input u_{des} and actual input u_{act}. The constraints of the program, which are known as *barrier constraints*, take the form $BC_i(x,u) = a_i(x)u + b_i(x) \geq 0$, $i \in \{1, ..., N\}$. The canonical ASIF controller is given below as:

ASIF-QP

$$u_{\text{act}}(x, u_{\text{des}}) = \text{argmin}_{u\in U}\|u - u_{\text{des}}\|_2^2 \tag{2.7}$$

$$\text{s.t.} \quad BC_i(x,u) = a_i(x)u + b_i(x) \geq 0, \qquad i = 1, ..., N \tag{2.8}$$

The ASIF-QP guarantees system safety with respect to an operational region $\mathcal{C}_{\text{S}}$ when the following two properties are satisfied by the barrier constraints:

- ϑ_1^{BC} : if (2.8) is satisfied then the system is forward invariant in some set $\mathcal{C}_{\text{S}} \subseteq \mathcal{C}_{\text{A}}$. *i.e.*

$$BC_i(x,u) \geq 0,\ \forall i \in \{1, ..., N\} \implies \mathcal{C}_{\text{S}} \text{ forward invariant} \tag{2.9}$$

  where $\mathcal{C}_{\text{S}} \subseteq \mathcal{C}_{\text{A}}$.

- $\vartheta_2^{\mathrm{BC}}$: it is possible to satisfy (2.8) from any state in the set $\mathcal{C}_{\mathrm{S}}$. *i.e.* $\forall x \in \mathcal{C}_{\mathrm{S}}$, and $\forall i \in \{1, \ldots, N\}$,

$$\sup_{u \in U}[BC_i(x, u)] \geq 0 \tag{2.10}$$

The first property ensures that the RTA mechanism will bound trajectories to the set $\mathcal{C}_{\mathrm{S}}$, as long as there exists a feasible input satisfying the barrier constraint. The second property ensures that for all states in $\mathcal{C}_{\mathrm{S}}$, it is possible to find a feasible input that satisfies the barrier constraint. Since quadratic programs can be solved in such a way that feasible solutions are found whenever they exist, guaranteeing the existence of a solution is sufficient for ensuring feasibility of the ASIF-QP. When both properties are satisfied, $\mathcal{C}_{\mathrm{S}}$ defines a safe set with respect to the ASIF. Moreover, the ASIF-QP is *minimally invasive* in the sense that the l^2 distance between u_{act} and u_{des} is minimized, subject to the barrier constraints.

As with the case of Simplex, implicit and explicit approaches emerge for ASIF based on how the safe operational region $\mathcal{C}_{\mathrm{S}}$ is identified. First, *explicit* ASIFs are studied. In this case, $\mathcal{C}_{\mathrm{S}}$ obtained by identifying a control invariant set explicitly as the super zero level-set of a smooth function. Next, in the case of *implicit* ASIFs, a control invariant set is identified implicitly through the trajectories of a backup control law. A more thorough review of explicit ASIF approaches is provided in [75] and implicit ASIF approaches in [76]. See also [77] for a review of both implicit and explicit ASIF approaches.

2.4.1 Barrier Constraints from an Explicitly Defined Safe Set

Consider a set $\mathcal{C}_{\mathrm{S}}$ that is defined as the super zero-level set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, *i.e.*

$$\mathcal{C}_{\mathrm{S}} = \{x \in X \,|\, h(x) \geq 0\} \tag{2.11}$$

$$\partial\mathcal{C}_{\mathrm{S}} = \{x \in \mathcal{C}_{\mathrm{S}} \,|\, h(x) = 0\}, \tag{2.12}$$

and suppose that 0 is a regular value of h; *i.e.* the gradient of h does not vanish on the boundary. The goal is to design a barrier constraint $BC(x, u)$ that satisfies $\vartheta_1^{\mathrm{BC}}$ and $\vartheta_2^{\mathrm{BC}}$, and consequently makes the set $\mathcal{C}_{\mathrm{S}}$ forward invariant under the ASIF-QP. One method for ensuring the forward invariance of $\mathcal{C}_{\mathrm{S}}$ involves checking a subtangentality condition over the boundary of $\mathcal{C}_{\mathrm{S}}$. Sometimes called Nagumo's condition, this result is as follows: if

$$L_f h(x) + L_g h(x) u(x) \geq 0 \tag{2.13}$$

holds for all $x \in \partial\mathcal{C}_{\mathrm{S}}$, then $\mathcal{C}_{\mathrm{S}}$ is forward invariant for the closed-loop dynamics of (2.6) under $u(x)$, where L_f, L_g in (2.13) denote Lie derivatives of h along f and g, respectively. Informally, (2.13) ensures that $\dot{h}(x) \geq 0$ for all $x \in \partial\mathcal{C}_{\mathrm{S}}$ and, thus, any smooth control law $u(x)$ satisfying (2.13) ensures the forward invariance of $\mathcal{C}_{\mathrm{S}}$ when applied to (2.6).

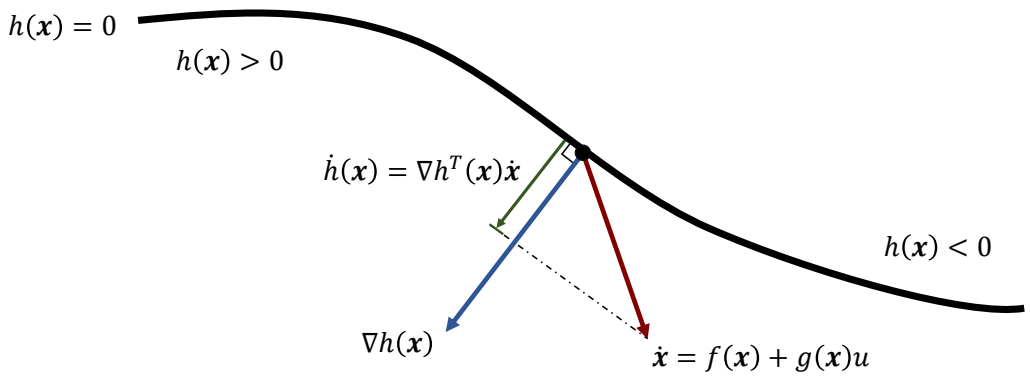


Figure 2.3: Barrier geometry for constraint function $h : \mathbb{R}^2 \to \mathbb{R}$.

The constraint (2.13) is not practical for use directly in an optimization framework as it is activated exactly on the boundary of $\mathcal{C}_{\mathrm{S}}$, which is a region without volume. The solution presented in [78] is to use similar condition that is active everywhere in $\mathcal{C}_{\mathrm{S}}$, and that includes a strengthening term which relaxes the constraint away from the boundary: for all $x \in \mathcal{C}_{\mathrm{S}}$

$$L_f h(x) + L_g h(x) u(x) + \alpha(h(x)) \geq 0 \tag{2.14}$$

where $\alpha : \mathbb{R} \to \mathbb{R}$ is an extended class κ_∞ function. A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is class κ_∞ if α is strictly increasing and $\alpha(0) = 0$. Since $\alpha(h(x)) = 0$ on the boundary, satisfaction of (2.14) implies satisfaction of (2.13). Hence, (2.14) implies $\mathcal{C}_\mathrm{S}$ is forward invariant for the closed-loop dynamics of (2.6) under $u(x)$. For this reason, an attractive method for constructing the ASIFs (2.7)–(2.8) is to form barrier constraints using

$$BC(x, u) = L_f h(x) + L_g h(x) u + \alpha(h(x)). \tag{2.15}$$

Applying the previous theoretical results, the resulting ASIF ensures the forward invariance of $\mathcal{C}_\mathrm{S}$ and therefore satisfies $\vartheta_1^{\mathrm{BC}}$, as discussed above.

In order to satisfy the requirement $\vartheta_2^{\mathrm{BC}}$, the functions h, α must be chosen in such a way that ensures that there always exists $u \in U$ satisfying the barrier constraint; equivalently,

$$\sup_{u \in U} [L_f h(x) + L_g h(x) u] \geq -\alpha(h(x)) \tag{2.16}$$

must hold for all $x \in \mathcal{C}_\mathrm{S}$. Though it is on occasion possible to obtain h, α analytically, satisfying this requirement is in general accomplished through the computation of a *control invariant* subset of $\mathcal{C}_\mathrm{A}$. Specifically, if $\mathcal{C}_\mathrm{S}$ is a control invariant set, then it is always possible to find a strengthening function $\alpha(x)$ such that $\vartheta_2^{\mathrm{BC}}$ is satisfied [61]. Alternatively, one may bypass choosing α altogether with a relaxed QP formulation (see [61], [77] for more details). Identifying a control invariant subset $\mathcal{C}_\mathrm{S} \subseteq \mathcal{C}_\mathrm{A}$ is the main challenge in implementing this method. Since $\mathcal{C}_\mathrm{S}$ defines the safe operational region for the RTA, it is desirable for this region to be as large as possible, with the largest possible control invariant subset being known as the viability kernel. Approximating viability kernels has been the subject of a wide variety of research. Approaches that have been proposed include: discretized solutions the Hamilton-Jacobi equations [79], sum-of-squares (SOS) optimization [59], and sampling [80]. See [77, 81] for a more extensive overview of techniques.

Example 2.4.1. Consider the double integrator system described by (1.11) and the constraint set (1.12). A control invariant set is given by the super zero level-set of $h(x) = -2x_1 - x_2^2$ and

given a class κ_∞ function α, the barrier constraint is $-2x_2(1+u) + \alpha(-2x_1 - x_2^2) \geq 0$. Note that $\forall x \in \mathcal{C}_\mathrm{S},\ \alpha(h(x)) \geq 0$ and $\sup_{u \in U}[L_f h(x) + L_g h(x) u] = 0$, hence the barrier constraint is viable. ■

Example 2.4.2. Consider the dynamics of a spacecraft in unconstrained rotation,

$$\dot{x} = J^{-1}(-x \times Jx) + J^{-1}u \tag{2.17}$$

where $x \in \mathbb{R}^3$ represents the angular velocity, $J = \mathrm{diag}(J_1, J_2, J_3) \in \mathbb{R}^{3\times 3}$ is a diagonal inertia matrix and $u \in [-1, 1]^3$. A control invariant set for this system is given by $\mathcal{C}_\mathrm{S} = \{x \in \mathbb{R}^3 \,|\, h(x) \geq 0\}$ with,

$$h(x) = K - x^T Jx \tag{2.18}$$

for any $K \in \mathbb{R}_+$. The fact that this set is control invariant is made apparent with the observation that the rotational kinetic energy $x^T Jx$ is constant when $u = 0$. Choosing $\alpha(x) = x$, the barrier constraint is $\nabla h^T(x)\dot{x} + h(x) \geq 0$, which simplifies to,

$$K - x^T Jx - 2x^T u \geq 0 \tag{2.19}$$

It is easy to show that this constraint is viable everywhere within $\mathcal{C}_\mathrm{S}$. Namely, $K - x^T Jx \geq 0$ everywhere in $\mathcal{C}_\mathrm{S}$, and one can always find $u \in U$ such that $-2x^T u \geq 0$.

■

If there exists an extended class κ_∞ function satisfying (2.16) for all $x \in X$ then $h(x)$ is said to be a *control barrier function* [75, 82, 83]. Barrier functions are were first developed in the context of nonlinear and hybrid system verification [84, 85]. The idea was first extended to the RTA problem with quadratic programs in [86], and it was developed more in [82]. In recent years, there has been a surge in interest in this topic. The theory has been extended to relax assumptions on smoothness [87], to discrete time [88] and stochastic [89] systems, and to enforce temporal logic specifications [90]. Likewise, the practicality of the approach is demonstrated in a variety of

Algorithm 3: Explicit Active Set Invariance Filter (EASIF)

input : Current State $x_{\text{curr}} \in X$
: Desired Input $u_{\text{des}} \in U$
output : Safe Control Input $u_{\text{act}} \in U$
predefined: Constraint Set $\mathcal{C}_{\text{A}} \subset X$
: Control Invariant Set $\mathcal{C}_{\text{S}} \subset \mathcal{C}_{\text{A}}$

1: **function** u_{act} =EASIF(x_{curr}, u_{des})
2: **solve:** (2.7) subject to the constraint (2.15).
3: **return** $u_{\text{act}}(x_{\text{curr}}, u_{\text{des}})$.

real-world applications, as shown in Table 2.1. It is important to note that in order for (2.15) to depend on the control input u, the constraint function $h(x)$ must have a relative degree of one; *i.e.* $L_g h(x) \neq 0$. Safety constraint functions with higher relative degrees are addressed in the literature on exponential control barrier functions [91, 75].

Example 2.4.3. Consider the system $\ddot{x} = u$ with $u \in U = (-\infty, \infty)$ and the constraint function $h(x) = -x_1$. Note that while actuation is unlimited in this case, it is not possible to enforce the barrier constraint $L_f h(x) + L_g h(x) u \geq \alpha(h(x))$ as $L_g h(x) = 0$. ■

Case Study: The Robotarium, An RTA Enabled Remote-Access Swarm Robotics Testbed

The Robotarium [65, 66, 67, 68] is a remotely accessible swarm robotics testbed with the stated mission to *democratize robotics by providing remote access to a state-of-the-art multi-robot research facility*. The lab consists of a 12ft×14ft custom arena and a number of differential drive robots, known as GritsBots. The testbed exhibits features such as Vicon-based real time motion-capture, wireless inductive charging, video capture of experiments, and an above mounted projector that allows users to project time-varying environmental projections onto the testbed during experiments. Users may submit code to control the robots through MATLAB or Python scripts, that are submited through the Robotarium website.

The open-access nature of the lab presents unique challenges in terms of safety. Experiments must remain as faithful as possible to the user specified behavior while being provably safe, so

as to prevent damage to the lab equipment. Offline verification of the user code would be a prohibitively difficult and time consuming task for the operators of the lab, and would place an unnecessary burden of correctness on its users. Consequently, an RTA solution to safety is utilized, and user submitted algorithms are treated as black box functions that can be probed at run time. Specifically, control barrier function based quadratic programs (explicit ASIF) are used to prevent collisions between the various robots involved in an experiment. An explicit ASIF-based approach is attractive in this case as it is minimally invasive to the desired inputs and provides a smooth overriding behavior. Since the planning is conducted under a single-integrator dynamics model, explicit identification of viable sets is fully-tractable problem offline.

2.4.2 Barrier Constraints from an Implicitly Defined Safe Set

As discussed in the previous section, the barrier constraint is viable when it restricts the system to a control invariant set. If an explicit representation of a large control invariant subset of $\mathcal{C}_{\mathrm{A}}$ can be obtained, then with the appropriate choice of α, a valid barrier constraint is obtained through (2.15). The key idea behind the implicit approach to ASIF [61, 23, 76] is to filter with respect to an implicitly defined control invariant set. Specifically, given a backup control law u_{b} and a backup set $\mathcal{C}_{\mathrm{b}}$ that is invariant under u_{b}, one can develop barrier constraints that activate near the boundary of the safe backward image of $\mathcal{C}_{\mathrm{b}}$. The idea was first proposed in [61] and further developed in [23, 76]. A tutorial on the topic is provided in [69], and some notable applications are listed in Table 2.1. A key advantage to this approach is that it does not require the computation of a large control invariant set.

Consider a smooth backup control law $u_{\mathrm{b}} : X \to U$ and suppose that the constraint set is given by

$$\mathcal{C}_{\mathrm{A}} := \{x \in X \,|\, \varphi(x) \geq 0\}, \tag{2.20}$$

and a backup set $\mathcal{C}_{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}}$ is given by

$$\mathcal{C}_{\mathrm{b}} = \{x \in X \,|\, h_{\mathrm{b}}(x) \geq 0\}. \tag{2.21}$$

Now considering an integration horizon of $[0, T_\mathrm{b}]$, the safe backward image of $\mathcal{C}_\mathrm{b}$ is

$$\mathcal{C}_\mathrm{S} = \{x \in \mathbb{R}^n \,|\, \forall t \in [0, T_\mathrm{b}],\ \varphi(\phi^{u_\mathrm{b}}(t; x)) \geq 0 \ \wedge\ h_\mathrm{b}(\phi^{u_\mathrm{b}}(T_\mathrm{b}; x)) \geq 0\}. \tag{2.22}$$

where $\phi^{u_\mathrm{b}}(t; x)$ is the flow of (2.6) under u_b, evaluated t seconds from x. Given an extended class κ_∞ function α, it can be shown that the following constraints are sufficient for invariance in $\mathcal{C}_\mathrm{S}$ [22]:

$$\frac{\mathrm{d}\varphi(t_\mathrm{b}; x))}{\mathrm{d}t} + \alpha(\varphi(\phi(t_\mathrm{b}; x))) \geq 0 \tag{2.23}$$

$$\frac{\mathrm{d}h_\mathrm{b}(\phi(T_\mathrm{b}; x))}{\mathrm{d}t} + \alpha(h_\mathrm{b}(\phi(T_\mathrm{b}; x)) \geq 0 \tag{2.24}$$

for all $t_\mathrm{b} \in [0, T_\mathrm{b}]$. The constraint (2.23) enforces that the points along the backup trajectory stay in the constraint space $\mathcal{C}_\mathrm{A}$, and (2.24) enforces that the endpoint of the backup trajectory stay in $\mathcal{C}_\mathrm{b}$. Expanding the derivative terms [22] yields the implicit ASIF barrier constraints:

$$\nabla\varphi(\phi^{u_b}(t_\mathrm{b}; x))D\phi^{u_b}(t_\mathrm{b}; x)[f(x) + g(x)u] + \alpha(\varphi(\phi^{u_b}(t_\mathrm{b}; x))) \geq 0 \tag{2.25}$$

$$\nabla h_\mathrm{b}(\phi^{u_b}(T_\mathrm{b}; x))D\phi^{u_b}(T_\mathrm{b}; x)[f(x) + g(x)u] + \alpha(h_\mathrm{b}(\phi^{u_b}(T_\mathrm{b}; x))) \geq 0. \tag{2.26}$$

for all $t_\mathrm{b} \in [0, T_\mathrm{b}]$. Note that since t_b lives in the interval $[0, T_\mathrm{b}]$, (2.25) represents an uncountable number of constraints. In practice one may approximate the constrained set of states by numerically integrating (2.6) forward under u_b and evaluating $\phi^{u_\mathrm{b}}(t; x)$ at discrete times $t_\mathrm{b} \in \{0, t_1, ..., t_{\mathrm{N}-1}, T_\mathrm{b}\}$. It is shown in [22] how the finite set of intermediate constraints may be tightened in such a way that is sufficient for satisfying the constraints with an infinite number of trajectory points. One may compute $D\phi^{u_\mathrm{b}}(t_\mathrm{b}; x)$ by integrating along with (2.6) a *sensitivity* matrix $Q(t_\mathrm{b}, x) = D\phi^{u_\mathrm{b}}(t_\mathrm{b}; x)$. As explained in [92], $Q(t_\mathrm{b}, x)$ is obtained as the solution to the following differential equation,

$$\frac{\mathrm{d}Q(t_\mathrm{b}, x)}{\mathrm{d}t_\mathrm{b}} = Df_\mathrm{cl}(\phi^{u_\mathrm{b}}(t_\mathrm{b}; x))Q(t_\mathrm{b}, x) \tag{2.27}$$

Algorithm 4: Implicit Active Set Invariance Filter (IASIF)

input : Current State $x_{\text{curr}} \in X$
: Desired Input $u_{\text{des}} \in U$
output : Safe Control Input $u_{\text{act}} \in U$
predefined: Constraint Set $\mathcal{C}_{\text{A}} \subset X$
: Invariant Backup Set $\mathcal{C}_{\text{b}} \subseteq \mathcal{C}_{\text{A}}$
: Smooth Backup Control Law $u_{\text{b}} : X \to U$.

function u_{act} =IASIF(x_{curr}, u_{des})
  Compute: $\phi^{u_{\text{b}}}(t_{\text{b}}; x_{\text{curr}}) \quad \forall t_{\text{b}} \in \{0, t_1, ..., t_{\text{N}-1}, T_{\text{b}}\}$ by integrating (2.6) under u_{b}
  Evaluate: $Df_{\text{cl}}(\phi^{u_{\text{b}}}(t_{\text{b}}; x_{\text{curr}})) \quad \forall t_{\text{b}} \in \{0, t_1, ..., t_{\text{N}-1}, T_{\text{b}}\}$
  Compute: $D\phi^{u_{\text{b}}}(t_{\text{b}}, x_{\text{curr}}) \quad \forall t_{\text{b}} \in \{0, t_1, ..., t_{\text{N}-1}, T_{\text{b}}\}$ by integrating (2.27)
  Solve: (2.7) subject to the constraints (2.25)-(2.26)
  return $u_{\text{act}}(x_{\text{curr}}, u_{\text{des}})$.

with $Q(0, x) = I_{n\times n}$ and where $Df_{\text{cl}}(\phi^{u_{\text{b}}}(t_{\text{b}}; x))$ is the Jacobian of the closed-loop dynamics of (2.6) under the control law $u_{\text{b}}(x)$, evaluated at $\phi^{u_{\text{b}}}(t_{\text{b}}; x)$. In order for this term to be defined, it is necessary that the control law u_{b} be smooth. As such, it is often necessary to consider smooth saturation functions for bounding the control law to the admissible domain U.

Example 2.4.4. Considered here is finite-time safety (collision avoidance) for two vehicles having combined dynamics,

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -b_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.28}$$

where x_1, x_3 denote positions and x_2, x_3 denote velocities for the vehicles, $b_1 = 0.1$ and $b_2 = 0.25$. The safety constraint is collision avoidance $\varphi(x) = x_3 - x_1$ and the constraint set is $\mathcal{C}_{\text{A}} = \{x \in \mathbb{R}^4 \,|\, \varphi(x) \geq 0\}$. To simplify the analysis, finite-time safety is considered, with the endpoint constraint (2.26) being omitted. The backup controller $u_{\text{b}} = [-1, 1]^T$ is integrated over a 10 s horizon and $\alpha(x) = 2x$. Figure 2.4 shows the result of a simulation with $u_{\text{des}}(t) = [1 - \exp(-0.1\,t),\, \exp(-0.25\,t) - 1]^T$. ■

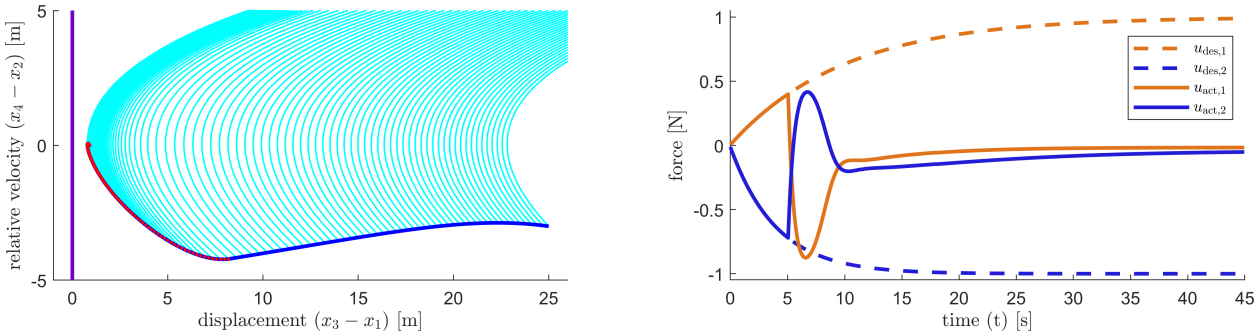


Figure 2.4: Simulation of the implicit ASIF algorithm for cooperative collision avoidance of two vehicles.

Example 2.4.5. Consider the system given by (2.17) and the constraint set $\mathcal{C}_{\text{A}} = \{x \in \mathbb{R}^3 \,|\, \varphi(x) \geq 0\}$ with,

$$\varphi(x) = \omega_{\max}^2 - x_1^2 - x_2^2 - x_3^3 \tag{2.29}$$

where $\omega_{\max} \in \mathbb{R}_{>0}$ represents the maximum allowable angular speed. The backup control law

$$u_{\text{b}} = \tanh((x \times Jx) - k_{\text{d}} Jx) \tag{2.30}$$

with $k_{\text{d}} > 0$ is bounded to $[-1, 1]$ and stabilizes system (2.17) to the origin. Furthermore, it can be shown that $\mathcal{C}_{\text{b}} = \{x \in \mathbb{R}^3 \,|\, K - x^T Jx \geq 0\}$ is invariant under $u_{\text{b}}(x)$ (see from previous example that this is invariant under the control law $u(x) \equiv 0$). The largest ellipsoid of this form that is contained in $\mathcal{C}_{\text{A}}$ is obtained by choosing $K = \omega_{\max}^2 / \min(J_1, J_2, J_3)$.

Figure 2.5 shows an illustration of the filtered trajectory under the (unsafe) primary controller $u_{\text{d}}(x) = [\sin(\frac{t}{2}), \sin(\frac{t}{2} - \frac{\pi}{4}), \sin(\frac{t}{4} + \frac{\pi}{4})]^T$ and with $\omega_{\max} = 1\,\text{m/s}$, $k_{\text{d}} = 1$, $J_1 = 12\,\text{kg}\,\text{m}^2$, $J_2 = 12\,\text{kg}\,\text{m}^2$, $J_3 = 5\,\text{kg}\,\text{m}^2$ and the backup horizon $t_{\text{b}} \in \{0, 0.05, ..., 3\}$. The trajectory is blue where $u_{\text{des}} = u_{\text{act}}$ and red where the RTA mechanism is active. ■

Application: Neural Network Approximation of Optimal Backup Controller

Here an application of the implicit ASIF algorithm is considered for a neural network controller. Consider a planar model of the two wheeled inverted pendulum (commonly referred to as a Seg-

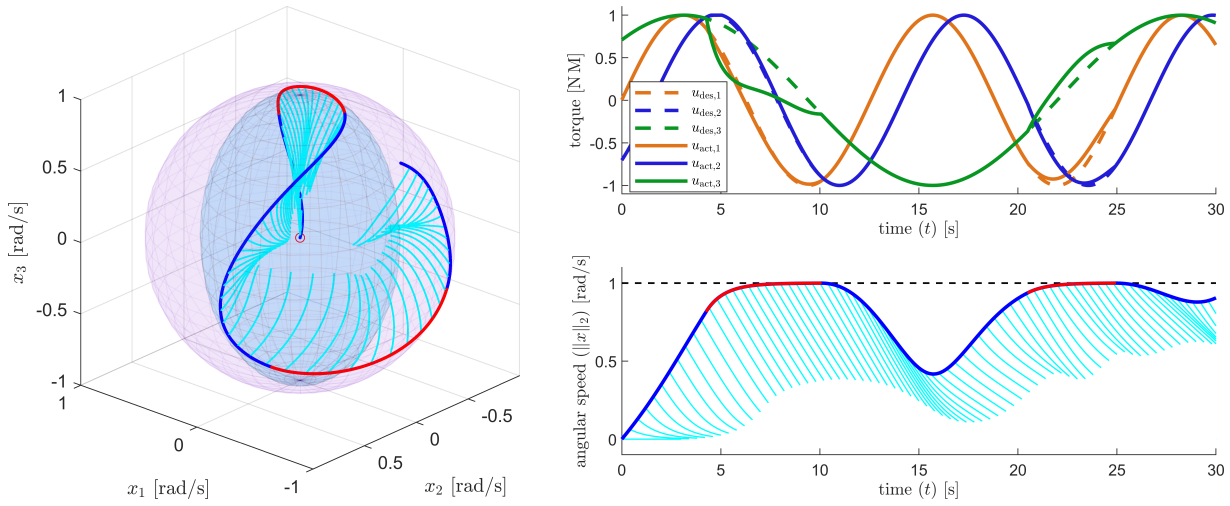


Figure 2.5: Simulation of implicit ASIF algorithm for spacecraft angular velocity dynamics. The constraint set $\mathcal{C}_{\text{A}}$ is represented by the purple sphere, the backup set $\mathcal{C}_{\text{b}}$ is represented by the blue ellipsoid. The surface of this ellipsoid has constant kinetic energy. The bottom left plot projects this information into normed space, and the top left plot shows the desired and actual control values.

way) shown in Figure 2.6. The state vector is composed of the following variables: position (p), velocity ($\dot{p}$), pitch angle (ψ) and pitch rate ($\dot{\psi}$), and the input (u) is taken as the voltage applied to the motors. The equations are derived using classical Lagrangian mechanics assuming no-slip between the ground and the wheels and using a first-order model of a DC motor. The resulting formulation is control affine.

A backup controller is derived from solutions to the following optimal control problem, stated in continuous time:

Optimal Control Problem (OCP)

$$\begin{aligned} u^*(x) = \underset{u \in U}{\text{argmin}} \quad & t_f + \textstyle\int_0^{t_f} x^T Q x + R u^2 \mathrm{d}t \\ \text{s.t.} \quad & \dot{x} = f(x) + g(x)u \\ & x(0) = x_0,\ x(t_f) = 0 \\ & x \in \mathcal{C}_{\text{A}} \end{aligned} \tag{2.31}$$

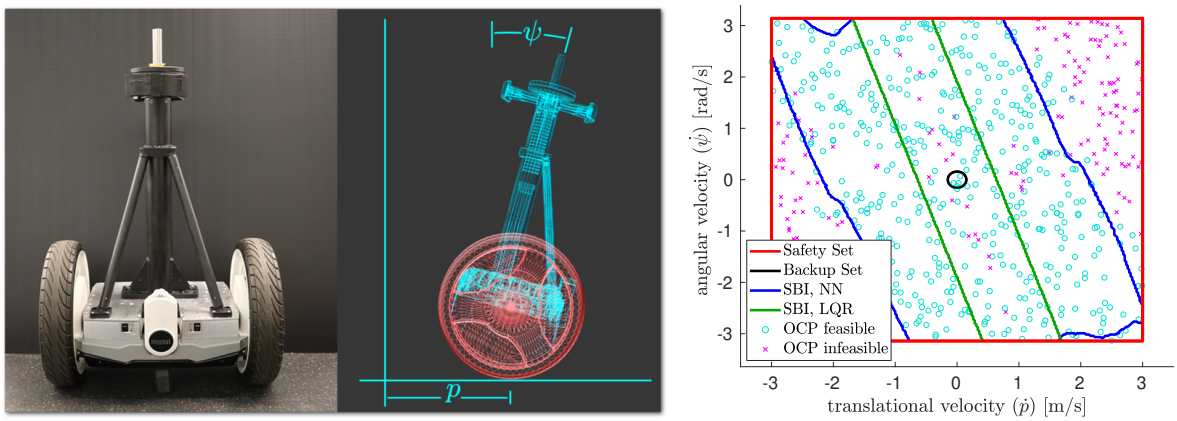


Figure 2.6: Segway vehicle (left); Comparison of the safe backward images (SBIs) of $\mathcal{C}_{\mathrm{b}}$ when using an LQR, solving the optimal control problem (OCP), and using a neural network regression of the OCP (right). The *safety set* here refers to the constraint space $\mathcal{C}_{\mathrm{A}}$.

where the state and input cost matrices are given values $Q = 0.02I_4$,and $R = 0.01$ respectively, and the input is bounded to the range $U = [-20, 20]$. The goal of this controller is to drive the vehicle to the origin while remaining in the constraint set,

$$\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R}^4 \mid |p| \leq 3, |\dot{p}| \leq 3, |\psi| \leq \tfrac{\pi}{4}, |\dot{\psi}| \leq \pi\}. \tag{2.32}$$

Note that penalizing the final time t_f conditions the solution to reach its goal under shorter time horizon. However, the numerical solution to purely time optimal formulation results in highly non-smooth output signals that often border constraint boundaries. As such, state and input penalties are added to the cost in order to make the solutions more amenable to functional approximation.

The OCP (2.31) is solved using GPOPS-II [93] software with the "hp-PattersonRao" mesh method at a tolerance of 0.01. Initial states for each trajectory are generated randomly within the bounds of $\mathcal{C}_{\mathrm{A}}$. A total of 816 trajectories are considered, comprising approximately 49000 trajectory points. The trajectory and control data are used to train a network $\mathcal{N}(x) : \mathbb{R}^4 \to \mathbb{R}$ consisting of a single hidden layer with 100 nodes. Hyperbolic tangent sigmoid (tansig) activation functions are used in this layer. The training process consists of 280 epochs with the "Levenberg–Marquardt" training algorithm, resulting in a mean squared error (MSE) of 2.96.

Sampled initial conditions are taken to characterize a level slice of the SBI for (i) the computed solution to the OCP, (ii) the neural network approximation $\mathcal{N}$, and (iii) an LQR controller with state and input cost matrices equal to those defined in (2.31) (see Figure 2.6). In this scenario, the initial position and angle of the vehicle are fixed at zero $x, \psi = 0$, and a set of initial velocity perturbations $\dot{x_0}, \dot{\psi_0}$ are given in the bounds of $\mathcal{C}_{\mathrm{A}}$. The backup set is chosen as $\mathcal{C}_{\mathrm{b}} = \{x \in \mathbb{R}^4 \,|\, h_{\mathrm{b}}(x) \geq 0\}$ with

$$h_{\mathrm{b}} := 0.1^2 - (p/3)^2 - (\dot{p}/3)^2 - (4\psi/\pi)^2 - (\dot{\psi}/\pi)^2, \tag{2.33}$$

and the controller is integrated over a horizon of $T = 5$ s.

The SBI for the LQR and neural network controllers are estimated by considering a 250×250 grid of initial velocity perturbations. The SBI boundaries indicated in Figure 2.6 separate the sets of sampled initial conditions for which a safe backup policies could always be found (interior) from the sets of sampled initial conditions where safe backup policies were never found (exterior). For the LQR controller, 21.0% of the samples in fall into the SBI. The neural network controller extends this to cover 67.3% of the sample points. The increased performance of the neural network can be attributed to both its suitability to the nonlinear dynamics, and the explicit consideration of state constraints in the training process.

One may also compare the performance of the approximated controller to the returned solution to the OCP directly. Figure 2.6 shows the result of a number of GPOPS-II evaluations corresponding to a set of uniformly distributed initial conditions in this domain. It is apparent that many feasible points extend beyond the SBI of $\mathcal{N}$, indicating that this region could be extended with more data and a larger network. Infeasible outputs arise when either the program fails to return a solution, or experiences numerical issues, resulting in a solution that violates one of the constraints. Feasible solutions are necessarily safe by definition of (2.31). In contrast to the controllers previously considered, the numerical solution to the OCP does not appear to have a clear boundary for its SBI. Solutions may fail in any subset of the domain. The neural network approximation excludes these failed trials, which effectively regularizes the output of the solver. Hence in addition to providing a more scalable approach, the learned control strategy is seen to demonstrate a

comparative improvement in numerical reliability over its operational domain.

2.4.3 Additional Design Considerations for ASIF

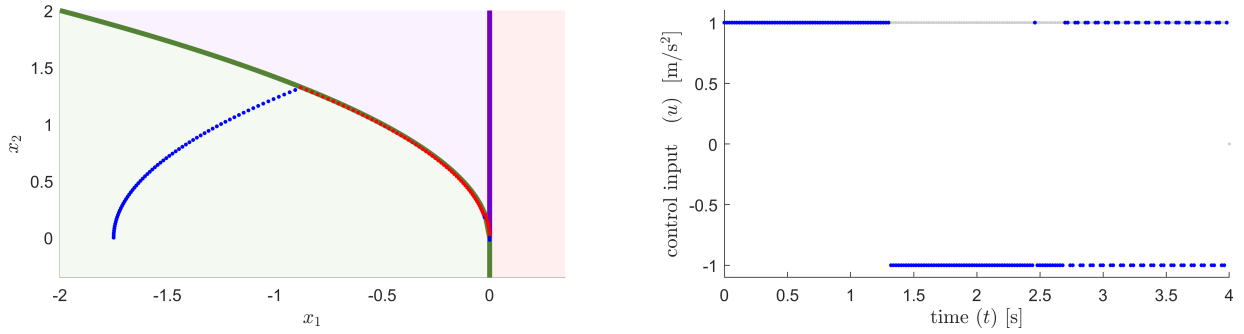
Recall that when the properties $\vartheta_1^{\mathrm{BC}}$ and $\vartheta_2^{\mathrm{BC}}$ are satisfied, a set $\mathcal{C}_{\mathrm{S}}$ is forward invariant under the RTA control law u_{act} given by the ASIF-QP. While certain guarantees exist under the assumptions of the model, and indeed the formulations presented exhibit some robustness properties, it is desirable in practice to consider the behavior of the system in the case where the state departs from $\mathcal{C}_{\mathrm{S}}$. A simple solution is to make $\mathcal{C}_{\mathrm{S}}$ a locally attractive set by switching to a stabilizing backup controller whenever $x \notin \mathcal{C}_{\mathrm{S}}$. Alternatively, for explicit ASIF, *if h is a concave function*, then the barrier constraint will cause u_{act} to attract solutions to the interior of $\mathcal{C}_{\mathrm{S}}$ whenever the ASIF-QP is feasible. A backup controller should be present for the case where a feasible solution is not found. While outside of the scope of this work, it is interesting to note that attracting solutions outside of $\mathcal{C}_{\mathrm{S}}$ to $\mathcal{C}_{\mathrm{S}}$ doesn't necessarily produce to the best response of the physical system. For instance Chapter 5 presents a method for minimizing a function of *damage* in the presence of an inevitable collision (see also [94]).

2.5 Discussion and Comparison of Approaches on Double Integrator System

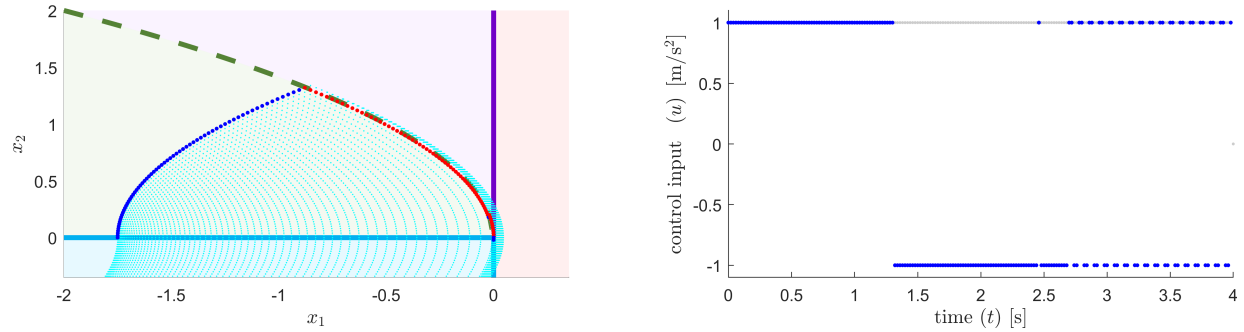
Figure 2.7 shows a comparison of the four algorithms presented in this chapter on the double integrator system given by (1.11) with $u \in [-1, 1]$, $\mathcal{C}_{\mathrm{A}} = \{x \in \mathbb{R}^2 \mid -x_1 \geq 0\}$, and a controller update period of 10 Hz. The simulations begin at state $x(0) = [-1.75, 0]^T$ and have desired input $u_{\mathrm{des}} = 1$. Here it can be seen that both Simplex-based algorithms exhibit near identical behavior, and likewise both ASIF algorithms exhibit similar behavior. In spite of the implicit methods not having explicit knowledge of the boundary, all cases exhibit similar operational regions. Note that while the actual trajectory remains safe, the backup trajectory under the implicit Simplex filter violates the safety constraint. This is by design, as the trajectories are integrated forward after taking a simulated *probe step*. The ASIF approaches are slightly more conservative due to the presence of the α term in the barrier constraints. This function may be modified to shape the

response.

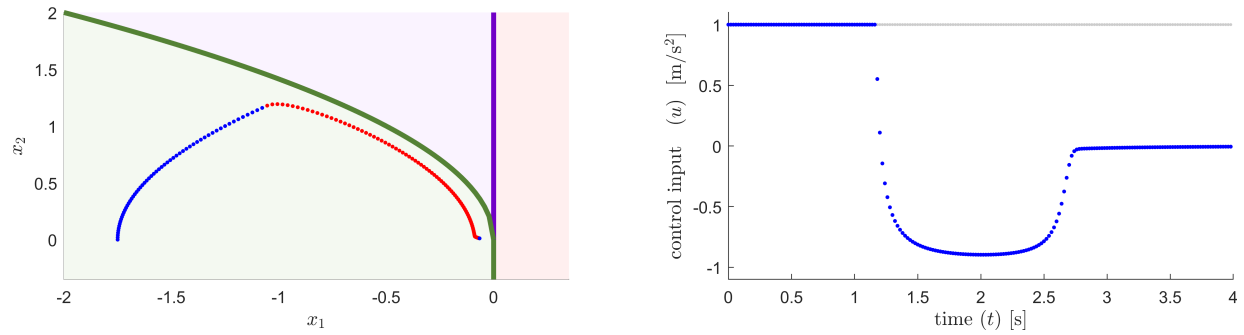
While the Simplex and ASIF methods are most distinctive in terms of behavior, the implicit and explicit methods are most distinctive in terms of requirements. For instance, the key challenge in using the implicit methods is the design of the backup controller and the identification of the backup set. Once these have been identified, it is relatively straightforward to implement either approach. Likewise, the key challenge in implementing the explicit algorithms is identifying a large invariant set. The explicit ASIF approach is the only algorithm that does not require any backup controller to be defined.



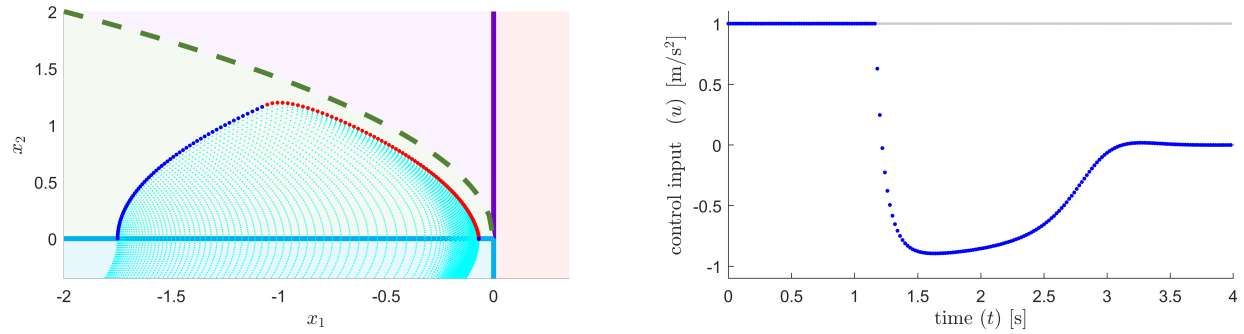
(a) Explicit Simplex filter (RBSF)



(b) Implicit Simplex filter (SBSF)



(c) Explicit active set invariance filter (EASIF)



(d) Implicit active set invariance filter (IASIF)

Figure 2.7: Comparison of algorithms on double integrator system (1.11) with constraint set (1.12) and $U = [-1, 1]$.

CHAPTER 3

SAFE AUTONOMOUS RENDEZVOUS PROXIMITY OPERATIONS AND DOCKING

Autonomous rendezvous, proximity operations, and docking (ARPOD) are key enablers of missions such as satellite servicing, active debris removal, and in-space assembly. However, errors in the control and estimation systems, or failures to account for off-nominal conditions may result in catastrophic collisions between spacecraft. Safety may potentially be preserved in these cases by switching to a safety-driven backup system. This chapter develops such a system, with guidance, control, and estimation schemes designed to safely place an active chaser spacecraft in a parking orbit around a passive target spacecraft. Natural motion trajectories are considered to identify a set of passively safe parking orbits under Clohessy-Wiltshire-Hill (CWH) dynamics, and a Mixed Integer Programming (MIP) formulation is developed to find the optimal transfer trajectories to this set. The practicality of the estimation and control schemes is demonstrated though simulated case studies. The guidance algorithm is integrated into a run time assurance (RTA) framework, which allows real-time enforcement of the safety constraints in a least-intrusive fashion.

3.1 Motivation

In an increasingly crowded space environment, ARPOD operations are envisioned to be integral to the design of future satellite servicing, active debris removal, and in-space assembly missions. However, one of the greatest risks in ARPOD missions is the potential for a collision between the active "chaser" satellite, and the passive "target" satellite. The severity of this risk is demonstrated by several historic collisions that occurred during rendezvous, proximity operations, and docking [95, 96, 97, 98].

This research proposes an optimal control technique for safe transfer maneuvers to a safely defined subset of *elliptical natural motion trajectories* (eNMTs), referred to as parking orbits. *Closed natural motion trajectories* (cNMTs), which include eNMTs as an instance, are periodic

(*i.e.* closed), zero-thrust trajectories that occur under linearized relative motion dynamics. They are attractive for collision avoidance because they are *passively invariant*; *i.e.* once a spacecraft enters a cNMT, it will stay on that trajectory, without the need to consume fuel, until disturbances such as drag or solar radiation pressure accumulate. The main contribution of this chapter is the development of a MIP formulation to plan safety-constrained maneuvers for an active chaser spacecraft to set of safe eNMTs around a passive target spacecraft. In addition, lower-level controllers are constructed to regulate to the planned trajectory, and to stabilize the system to the parking orbit set.

One important application of this approach is for it to be used as a backup control system in an RTA framework. RTA mechanisms typically feature a monitor that evaluates whether the primary control system is commanding unsafe control actions, and a backup control system that steers the system back to safety when unsafe behavior is detected. For example, switching to the backup system may occur when,

1. a human operator watching the system's performance is uncomfortable with the ARPOD controller output and activates the recovery maneuver,

2. the onboard RTA monitor detects the ARPOD controller is commanding an unsafe maneuver and triggers a recovery response, or

3. there is a fault in the ARPOD controller that triggers a recovery response.

Maneuvering the spacecraft to an eNMT parking orbit allows human operators on the ground to investigate the off-nominal condition while the chaser spacecraft is safely "parked" close to the inactive target spacecraft. Alternatively the backup controller may be activated only as necessary to prevent the chaser spacecraft from reaching unsafe states, as shown in the example in Section 3.6.

3.2 Related Work

MIP has been shown to be an effective framework for finding optimal trajectories under various types of operational and safety constraints [8, 25, 99, 4]. In the spacecraft domain, MIP formulations exist for fuel optimal guidance under plume impingement and avoidance constraints [100],

complex proximity operations [101], constrained attitude guidance [102], and planning involving low-speed bouncing maneuvers [94].

The backup control strategy in this chapter is an optimization-based collision avoidance algorithm for close proximity spacecraft operations. Several optimization techniques have also been proposed for collision avoidance strategies, both at orbital and relative motion scales. The NLP2 algorithm [103] minimizes delta-V and can be configured to include an optional miss distance constraint with uncertainties in individual or joint covariances. Bombardelli and Henando-Ayuso [104] developed an approach optimized eigenvalue problem that conducts an impulse collision avoidance a few orbits prior to the predicted collision to minimize total ΔV. Other optimal control techniques have examined the use of drag and solar radiation pressure as control inputs for collision avoidance [105].

At the relative motion dynamics scale, robust control approaches have used model predictive control and an extended command governor [106], as well as safe positively invariant sets [107]. Stochastic reachability has also been applied to characterize the initial set of states where spacecraft rendezvous and docking maneuvers could be performed safely [108]. This work compares particle (Monte Carlo) approximations [109, 110] to a convex, overapproximated, reachable set [111, 112]. While not specifically designed for collision avoidance, recent developments in spacecraft relative motion control and planning [113, 114, 115, 116, 106] inspired the eNMT-based collision avoidance trajectories in this chapter.

3.3 Preliminaries

3.3.1 Hill's Frame

The dynamics in this research are based on the rectangular, satellite-centered, relative motion, Hill's reference frame. This was initially developed in the 1870s to describe the relative orbit between two bodies in orbit around a central body [117]. Hill's frame is also sometimes referred to as the local-vertical local-horizontal (LVLH) or radial-tangential normal (RTN) frame when centered on the Earth. Hill's frame, as depicted in Figure 3.1, is centered on the "target" (also

called "chief") satellite, with:

- x-axis $\hat{x}$ (or $\hat{e}_R$ for "radial direction") that points outwards from the planet's center ($\hat{x} = \hat{e}_R = \frac{r_e}{r_e}$),
- y-axis $\hat{y}$ (or $\hat{e}_T$ for "tangential direction") points in the direction of the velocity vector ($\hat{y} = \hat{e}_T = \hat{e}_N \times \hat{e}_R$) of the target satellite, and
- z-axis $\hat{z}$ (or $\hat{e}_N$ for "normal direction") points orthogonally out of the orbital plane ($\hat{z} = \hat{e}_N = \frac{r_e \times v_e}{||r_e \times v_e||}$, where r_e and v_e are the position and velocity vectors of the target satellite).

The x and y axes define the orbital plane of the satellite at the center of Hill's frame, while any motion along z is outside the orbital plane. In Figure 3.1, v_e is the velocity vector of the target spacecraft orbit, R_e is the vector from the planet's center to the chaser spacecraft, and r_e is a vector from the planet's center and to the location of the target satellite. The relative position of the chaser spacecraft is $r = r_x\hat{x} + r_y\hat{y} + r_x\hat{z}$, and the relative velocity is $v = v_x\hat{x} + v_y\hat{y} + v_z\hat{z}$.

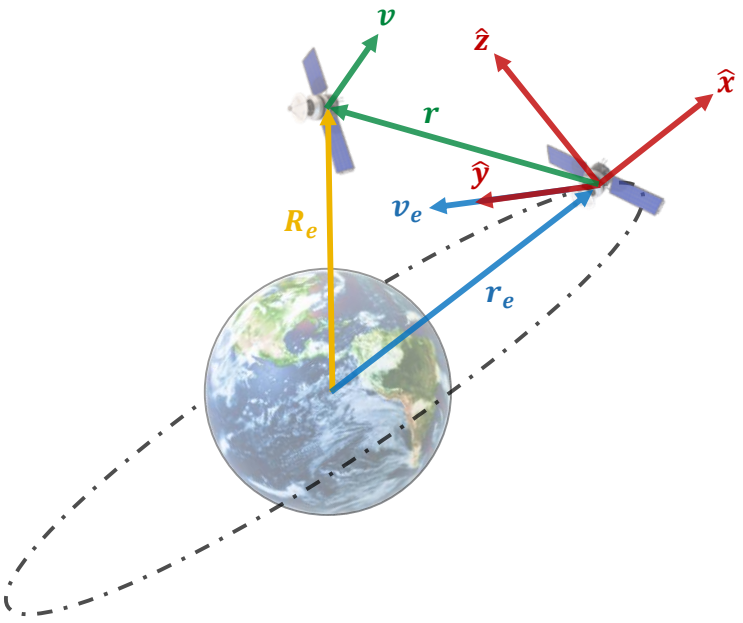


Figure 3.1: Hill's reference frame centered on a target spacecraft and used to describe the relative motion of a chaser spacecraft conducting ARPOD.

3.3.2 Clohessy-Wiltshire Equations

The orbital motion of a spacecraft is described by Eq. 3.1, where μ is the gravitational parameter ($(3.986004418 \pm 0.000000008) \times 10^{14} \frac{\mathrm{m}^3}{\mathrm{s}^2}$ for Earth), R_e is a vector from the planet center to the satellite ($R_e = r_e + r$), m_c is the mass of the satellite, and $F = F_{\mathrm{x}}\hat{x} + F_{\mathrm{y}}\hat{y} + F_{\mathrm{z}}\hat{z}$ is a vector of external forces that may include thrust as well as disturbances.

$$\ddot{R}_e = -\mu \frac{R_e}{|R_e|^3} + \frac{1}{m_c} F \tag{3.1}$$

These dynamics can be used to derive and linearize the relative motion dynamics by assuming the satellites are spherical and roughly equivalent in mass, that masses of the planet and the satellites are constant, that the mass of the satellites are significantly smaller than the mass of the planet, that gravity is the only force acting on the satellites and planet, that the spacecraft are in circular orbit, and that the relative distance between the spacecraft is much smaller than the distance from either spacecraft to the center of the planet ($r << R_e$) [31]. These linearized dynamics are named after Clohessy and Wiltshire from their 1960 paper [31] and are defined in Hill's frame as:

$$\begin{aligned} \dot{v}_{\mathrm{x}} &= 2nv_{\mathrm{y}} + 3n^2 r_{\mathrm{x}} + \frac{F_{\mathrm{x}}}{m_c} \\ \dot{v}_{\mathrm{y}} &= -2nv_{\mathrm{x}} + \frac{F_{\mathrm{y}}}{m_c} \\ \dot{v}_{\mathrm{z}} &= -n^2 r_{\mathrm{z}} + \frac{F_{\mathrm{z}}}{m_c} \end{aligned} \tag{3.2}$$

where dotted variables indicate derivatives with respect to time; a is length of the semi-major axis of the target spacecraft's orbit; and n is satellite mean motion ($n = \sqrt{\mu/a^3}$). In this chapter, $n = 0.001027$ rad/s is used.

The external forces are assumed to be equal to the applied thrust $u = u_{\mathrm{x}}\hat{x} + u_{\mathrm{y}}\hat{y} + u_{\mathrm{z}}\hat{z}$, with u acting as the *control* variable. As evident by inspection of these equations, the in-plane dynamics (involving r_{x}, r_{y} and their derivatives) are decoupled from the out-of-plane dynamics (involving r_{z} and its derivatives). In addition, the in-plane dynamics are unstable with two eigenvalues at the

origin and two at $\pm nj$. The out-of-plane dynamics are stable with two eigenvalues at $\pm nj$. The in-plane dynamics are completely controllable from u_{y} but not controllable from u_{x}, and the out-of-plane dynamics are controllable from u_{z}. Given that the out-of-plane dynamics are decoupled and completely controllable with u_{z}, it is convenient to restrict focus to in-plane relative motion. In state space form, in-plane dynamics are described as:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.3}$$

with, $x = [r_{\mathrm{x}},\, r_{\mathrm{y}},\, v_{\mathrm{x}},\, v_{\mathrm{y}}]^T \in \mathcal{X} \subset \mathbb{R}^4$ being the state vector, $u = [u_{\mathrm{x}},\, u_{\mathrm{y}}]^T \in U \subset \mathbb{R}^2$ being the control vector, $t \in \mathbb{R}$ being the time, and

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m_c & 0 \\ 0 & 1/m_c \end{bmatrix}. \tag{3.4}$$

The guidance and control formulation in this chapter assumes the above dynamics with thrust limitations given by $U = [-u_{\max}, u_{\max}]^2$.

3.3.3 Natural Motion Trajectories

Natural Motion Trajectories (NMTs) are solutions to (3.2) (or (3.3)) with all applied forces equal to zero ($u \equiv 0$). Open NMTs may be a non-periodic line segment, helix (traveling ellipse), or spiral, while cNMTs may be ellipses (eNMTs), periodic line segments or stationary points. To create any cNMT, one must choose an initial state such that $v_{\mathrm{y}}(0) = -2nr_{\mathrm{x}}(0)$ where n is the mean motion of the circular orbital trajectory. In addition to this criteria, each cNMT has specific additional criteria.

- For a cNMT to be a *stationary point*, $r_{\mathrm{x}}(0) = r_{\mathrm{z}}(0) = v_{\mathrm{x}}(0) = v_{\mathrm{z}}(0) = 0$.
- To create a *periodic line segment*, $r_{\mathrm{x}}(0) = v_{\mathrm{x}}(0) = 0$, $r_{\mathrm{z}}(0) = c\sin(\psi)$, and $v_{\mathrm{z}}(0) =$

$nc\cos(\psi)$, where c is the magnitude of the oscillation (1/2 the length of the line segment) and n is the mean motion as previously defined.

- To create an eNMT, $v_{\mathrm{x}}(0) = \frac{n}{2} r_{\mathrm{y}}(0)$.

Each of these conditions assumes that the state variables fall within ranges for which the linearized dynamics developed in the previous section are valid.

3.3.4 Maximum and Absolute Value Assignments with Mixed Integer Programming

Recall from Section 1.4 that integer variables allow for the direct expression of first order logic over the constraints [118, 3]. This chapter employs a MIP formulation for constrained optimal guidance of the chaser spacecraft. For this purpose, it is convenient to review the below relations. Consider scalar variables $\bar{x}, x \in [x_{\min}, x_{\max}]$, integer variable $\zeta \in \{0, 1\}$ and constant $M \geq \max(x_{\min}, x_{\max})$. Them the assignment $\bar{x} = |x|$ is equivalent to the following set of constraints,

$$\begin{aligned} x \leq \bar{x} \quad &\wedge \quad -x \leq \bar{x} \\ x \leq M\zeta \quad &\wedge \quad -x \leq M(1-\zeta) \\ \bar{x} \leq x + M(1-\zeta) \quad &\wedge \quad \bar{x} \leq -x + M\zeta. \end{aligned} \tag{3.5}$$

A scalar variable $x_{\max}$ may be assigned the maximum value of two scalar variables in a similar fashion. Given $\zeta \in \{0, 1\}$, the assignment $x_3 = \max(x_1, x_2)$ is equivalent to the constraints,

$$\begin{aligned} x_1 \leq x_3 \quad &\wedge \quad x_2 \leq x_3 \\ x_1 - x_2 \leq M\zeta \quad &\wedge \quad x_2 - x_1 \leq M(1-\zeta) \\ x_3 \leq x_1 + M(1-\zeta) \quad &\wedge \quad x_3 \leq x_2 + M\zeta. \end{aligned} \tag{3.6}$$

The equivalency of the constraints in (3.5) and (3.6) to their respective assignments is evident in the expression of a truth table.

3.3.5 Passive Invariance

Intuitively, a control policy is safe if it prevents the system from visiting undesirable states for all time, whenever it is possible to do so. As described in Section 1.6, such a notion of safety can be formalized in terms of guarantees on set *invariance*. Let $\mathcal{C}_{\mathrm{A}}$ be the *constraint space*, which is the set of all states that satisfy the safety constraints. Then a system is safe with under a given control law if that control law renders a subset of the constraint space forward invariant. For spacecraft, it is important to consider a definition of safety that accounts for fuel usage. It is said that a spacecraft is *passively safe* if it is in a passively invariant set contained in $\mathcal{C}_{\mathrm{A}}$.

Definition 3.3.1 (passively invariant)**.** A closed set $\mathcal{S}$ is *passively invariant* for system (3.3) if the solution to (3.3) with $u \equiv 0$ satisfies, $x(t_0) \in \mathcal{S} \implies \forall t \geq t_0,\ x(t) \in \mathcal{S}$. $\triangle$

Note that if a safe finite-time trajectory can be found with an endpoint that lies in a safe passively invariant set, then safety can be guaranteed for all time using only the fuel required to complete the trajectory.

3.4 Safety Constraints

Three safety constraints are defined on the system based on the norms of the position vector $r := [r_{\mathrm{x}},\, r_{\mathrm{y}}]^T$ and velocity vector $v := [v_{\mathrm{x}},\, v_{\mathrm{y}}]^T$. First, the collision avoidance constraint,

$$\varphi_1(x) := \|r\| - r_{\min} \geq 0 \tag{3.7}$$

imposes the requirement that the chaser spacecraft avoid coming within a distance of $r_{\min}$ of the target spacecraft. Second, the proximity constraint,

$$\varphi_2(x) := r_{\max} - \|r\| \geq 0 \tag{3.8}$$

requires that the chaser spacecraft stay within a distance of $r_{\max}$ of the target. This distance may for example be chosen as the maximum distance where the linearization in Eq. (3.3) is considered

valid. Third, the constraint,

$$\varphi_3(x) := \kappa_1 + \kappa_2\|r\| - \|v\| \geq 0 \tag{3.9}$$

introduces a limit on the maximum safe relative speed between chaser and target, as a function of the separation distance. This chapter assumes values $\kappa_1 \geq 0$, and $\kappa_2 \geq 2n$. The safety constraints (3.7)-(3.9) together define the set of safe states,

$$\mathcal{C}_{\mathrm{S}} := \{x \in \mathbb{R}^4 \mid \varphi_i(x) \geq 0,\ i = 1, 2, 3\}. \tag{3.10}$$

The trajectory planning formulation considers the l^∞ norm case for the safe set definition. The constraints defining $\mathcal{C}_{\mathrm{S}}$ are represented visually in Figure 3.2 for $r_{\min} = 0.5$ km, $r_{\max} = 8$ km, $\kappa_1 = 1$ m/s, and $\kappa_2 = 2n\ \mathrm{s}^{-1}$.

3.4.1 Identification of Safe Parking Orbit Set

To construct the safe parking orbit set, the set of all eNMTs that are contained within the safe set $\mathcal{C}_{\mathrm{S}}$ is considered. Recall that the chaser spacecraft is on an eNMT centered at the origin exactly when the following constraints are satisfied,

$$\vartheta_1(x) := v_{\mathrm{x}} - \frac{n}{2}\, r_{\mathrm{y}} = 0, \tag{3.11}$$

$$\vartheta_2(x) := v_{\mathrm{y}} + 2n\, r_{\mathrm{x}} = 0. \tag{3.12}$$

Hence, the set of all states on eNMTs centered at the origin (target) is given by,

$$\mathcal{M} = \{x \in \mathbb{R}^4 \mid \vartheta_1(x) = \vartheta_2(x) = 0\}. \tag{3.13}$$

Since this set consists entirely of cNMTs, $\mathcal{M}$ is by nature a passively invariant set. Under the ideal assumptions of the model, if the chaser can reach this set, it can stay in the set indefinitely without using fuel. However, note that not all trajectories comprising $\mathcal{M}$ are safe. A safe subset of $\mathcal{M}$ —specifically, a passively invariant subset of $\mathcal{C}_{\mathrm{S}} \cap \mathcal{M}$— may be constructed by restricting this set to only include the states of eNMTs contained in $\mathcal{C}_{\mathrm{S}}$.

Recall that each eNMT has a period of $\tau = 2\pi/n$ and major axis equal to twice the minor axis. Consequently, individual eNMTs can be parameterized in terms of the semi-minor (or semi-major) axis. Let,

$$\vartheta_3(x) := r_{\mathrm{x}}^2 + \frac{r_{\mathrm{y}}^2}{4}, \tag{3.14}$$

then,

$$\eta(b) = \{x \in \mathcal{M} \mid \vartheta_3(x) = b^2\} \tag{3.15}$$

denotes the set of states occupied by an eNMT with semi-minor axis b. Projections of $\eta(b)$ in the r_{x}-r_{y} plane are level curves of ϑ_3. Projections in the v_{x}-v_{y} plane may be obtained similarly by substituting $r_{\mathrm{x}} = -(2n)^{-1}v_{\mathrm{y}}$, and $r_{\mathrm{y}} = (0.5n)^{-1}v_{\mathrm{x}}$ into Eq. (3.14). Various simulated projections of $\eta(b)$ are shown in Figure 3.2.

The trajectory corresponding to $\eta(b)$ is safe when,

$$\min_{x \in \eta(b)} \varphi_i(x) \geq 0, \quad i = 1, 2, 3. \tag{3.16}$$

It can be shown that,

$$\min_{x \in \eta(b)} \varphi_1(x) = \sqrt{\frac{4}{5}}\, b - r_{\min} \tag{3.17}$$

$$\min_{x \in \eta(b)} \varphi_2(x) = r_{\max} - 2b \tag{3.18}$$

$$\tag{3.19}$$

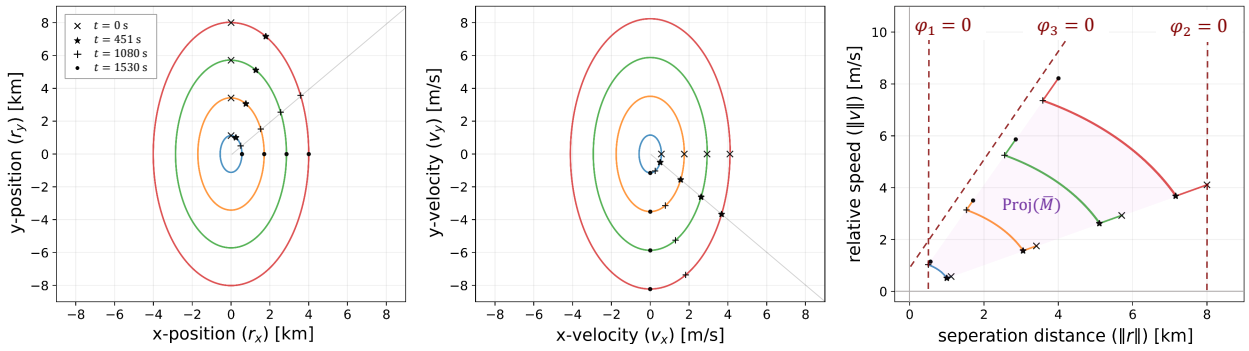


Figure 3.2: Projections of elliptical natural motion trajectories $\eta(b)$ with $b = 560\,\mathrm{m},\ 1706\,\mathrm{m},\ 2853\,\mathrm{m},\ 4000\,\mathrm{m}$, generated by simulating dynamics (3.3) over the horizon $\tau = 2\pi/n$ with initial conditions $x(0) = [0,\ 2b,\ nb,\ 0]^T$, and $u \equiv 0$.

and for all $\kappa_1 \geq 0$ and $\kappa_2 \geq 2n$,

$$\min_{x \in \eta(b)} \varphi_3(x) \geq 0. \tag{3.20}$$

Thus, the range of safe semi-minor axis values is $b \in \mathcal{B} = [b_{\min},\ b_{\max}]$ with

$$b_{\min} = \sqrt{\frac{5}{4}}\, r_{\min}, \quad b_{\max} = \frac{1}{2} r_{\max}. \tag{3.21}$$

The safe parking orbit set may be represented with these values as,

$$\bar{\mathcal{M}} := \bigcup_{b \in \mathcal{B}} \eta(b) = \{x \in \mathcal{M} \mid b_{\min}^2 \leq \vartheta_3(x) \leq b_{\max}^2\}. \tag{3.22}$$

Note that $\bar{\mathcal{M}} \subseteq \mathcal{C}_{\mathrm{S}}$ is passively invariant, as it is the union of cNMTs $\eta(b)$. It is defined by two equality constraints, and two inequality constraints. A projection of this set for $r_{\min} = 0.5$ km, $r_{\max} = 8$ km is shown in Figure 3.2.

3.5 Safety-Constrained Optimal Transfer Trajectories to Parking Orbit Set

This section formulates the problem of generating safe optimal transfer trajectories of a spacecraft from an initial state $x(0)$ to the safe parking orbit set $\bar{\mathcal{M}}$. It is shown that the dynamics of the spacecraft, actuation constraints, and safety constraints are all amenable to approximation with

piecewise affine constraints. In light of this, the trajectory optimization problem is posed as a MIP. A discrete time approximation of the dynamics (3.3) is considered over a horizon with sampled times t_i indexed by $i \in \{1, \ldots, \tau\}$. The period separating all points in the sample horizon is constant $\Delta t = t_{i+1} - t_i$, and the length of the horizon is $T = t_\tau - t_1$; the initial reference time is $t_1 = 0$. The state of the vehicle at the i^{th} time-step is represented by the variables $x_i = [r_{\mathrm{x},i},\, r_{\mathrm{y},i},\, v_{\mathrm{x},i},\, v_{\mathrm{y},i}]^T$, and control vector is denoted by $u_i = [u_{\mathrm{x},i},\, u_{\mathrm{y},i}]^T$. Safety is guaranteed over $t \in [0, T]$ with the requirement that $\forall i \in \{2, ..., \tau - 1\},\ x_i \in \mathcal{C}_\mathrm{S}$. Furthermore, safety is ensured $\forall t \geq T$ with the constraint $x_\tau \in \bar{\mathcal{M}}$.

Intermediate Point Constraints

Intermediate point constraints enforce that,

$$x_i \in \mathcal{C}_\mathrm{S}, \quad i = 2, ..., \tau - 1. \tag{3.23}$$

Recall that the safety set $\mathcal{C}_\mathrm{S}$ is defined by constraints (3.7)-(3.9), which are linear inequalities of the position and velocity norms. For the case where the l^∞ norm is used to define $\mathcal{C}_\mathrm{S}$, (3.23) may be represented *exactly* as a set of mixed integer constraints involving the state variables. This is apparent when one notes that variable assignments with both the maximum and absolute value operations can be represented by an equivalent set of mixed integer constraints (see Section 3.3.4). In this section, we represent assignments of these two operations directly with the understanding that the corresponding set of MIP constraints may be obtained by applying (3.5), (3.6). Note that direct assignments such as this are supported by some off-the-shelf optimization interfaces; *e.g.* the Python interface to Gurobi [11] is used for the results in this chapter.

Consider,

$$\bar{r}_{\mathrm{x},i} = \mathrm{abs}(r_{\mathrm{x},i}) \ \wedge \ \bar{r}_{\mathrm{y},i} = \mathrm{abs}(r_{\mathrm{y},i}) \tag{3.24}$$

$$\bar{v}_{\mathrm{x},i} = \mathrm{abs}(v_{\mathrm{x},i}) \ \wedge \ \bar{v}_{\mathrm{y},i} = \mathrm{abs}(v_{\mathrm{y},i}) \tag{3.25}$$

with $i = 2, ..., \tau - 1$. These constraints set the value of decision variables $\bar{r}_{\mathrm{x},i}, \bar{r}_{\mathrm{y},i}, \bar{v}_{\mathrm{x},i}, \bar{v}_{\mathrm{y},i}$ to be equal to the absolute values of the corresponding position and velocity components. Variables $r_{\infty,i}$, $v_{\infty,i}$ are set equal to the position and velocity norms respectively with,

$$r_{\infty,i} = \max(\bar{r}_{\mathrm{x},i}, \bar{r}_{\mathrm{x},i}) \tag{3.26}$$

$$v_{\infty,i} = \max(\bar{v}_{\mathrm{y},i}, \bar{v}_{\mathrm{y},i}) \tag{3.27}$$

where $i = 2, ..., \tau - 1$. The safety constraints are constructed with these variables as,

$$r_{\infty,i} - r_{\min} \geq 0 \tag{3.28}$$

$$r_{\min} - r_{\infty,i} \geq 0 \tag{3.29}$$

$$\kappa_1 + \kappa_2 r_{\infty,i} - v_{\infty,i} \geq 0 \tag{3.30}$$

with $i = 2, ..., \tau - 1$. The set of constraints (3.24)-(3.30) is equivalent to (3.23).

End Point Constraints

End point constraints enforce that,

$$x_i \in \bar{\mathcal{M}}, \quad i = \tau, \tag{3.31}$$

by restricting the state to a polytope contained in $\bar{\mathcal{M}}$. The constraints $\vartheta_1(x_\tau) = \vartheta_2(x_\tau) = 0$ are represented directly as,

$$v_{\mathrm{x},i} - \frac{n}{2} r_{\mathrm{y},i} = 0, \tag{3.32}$$

$$v_{\mathrm{y},i} + 2n r_{\mathrm{x},i} = 0. \tag{3.33}$$

where $i = \tau$. The nonlinear inequality constraints defining the outer bound on position $\vartheta_3(x_\tau) \leq b_{\max}^2$ and inner position bound $\vartheta_3(x_\tau) \geq b_{\min}^2$ are approximated conservatively using a finite number of linear inequality constraints. Consider an N_{Q} sided polygon $\mathcal{Q}$ inscribed in the ellipse

$\vartheta_3 = b_{\max}^2$, with vertices,

$$q_j = \begin{bmatrix} q_{\mathrm{x},j} \\ q_{\mathrm{y},j} \end{bmatrix} = \begin{bmatrix} b_{\max}\cos(2\pi j/N_{\mathrm{Q}}) \\ 2b_{\max}\sin(2\pi j/N_{\mathrm{Q}}). \end{bmatrix},$$

where $j = 1, ..., N_{\mathrm{Q}}$. Let,

$$\begin{aligned} \alpha_{\mathrm{x},j} &:= (q_{\mathrm{y},j} - q_{\mathrm{y},j+1}) \\ \alpha_{\mathrm{y},j} &:= (q_{\mathrm{x},j} - q_{\mathrm{x},j+1}) \\ \gamma_j &:= \alpha_{\mathrm{x},j} q_{\mathrm{x},j} + \alpha_{\mathrm{y},j} q_{\mathrm{y},j} \end{aligned}$$

then,

$$\bigwedge_{j=1,...,N_{\mathrm{Q}}} \alpha_{\mathrm{x},j} r_{\mathrm{x},i} + \alpha_{\mathrm{y},j} r_{\mathrm{y},i} \leq \gamma_j \tag{3.34}$$

with $i = \tau$ gives a sufficient condition for $\vartheta_3(x_\tau) \leq b_{\max}^2$ by ensuring that the position stays in $\mathcal{Q}$.

The constraint $\vartheta_3 \geq b_{\min}^2$ is enforced similarly by requiring that the position avoid the interior of a polygonal outer approximation of the ellipse defined by $\vartheta_3 = b_{\min}^2$. However, since $\vartheta_3 \geq b_{\min}^2$ is non-convex, approximating this constraint requires the introduction of integer variables. Consider an N_{P} sided polygon $\mathcal{P}$ with edges normal to the ellipse at intersection points,

$$p_j = \begin{bmatrix} p_{\mathrm{x},j} \\ p_{\mathrm{y},j} \end{bmatrix} = \begin{bmatrix} b_{\min}\cos(2\pi j/N_{\mathrm{P}}) \\ 2b_{\min}\sin(2\pi j/N_{\mathrm{P}}). \end{bmatrix},$$

where $j = 1, ..., N_{\mathrm{P}}$. Unit normal vectors to the edges are, $\hat{n}_{\mathrm{p},j} = [\hat{n}_{\mathrm{px},j}, \hat{n}_{\mathrm{py},j}]^T := \nabla\vartheta_3(r)|_{p_{\mathrm{j}}}$. Letting $r_i := [r_{\mathrm{x}_i}, r_{\mathrm{y}_i}]^T$, and $\zeta_{i,j} \in \{0, 1\}$, the constraints,

$$\bigwedge_{j=1,...,N_{\mathrm{Q}}} \hat{n}_{\mathrm{p},j}^T (r_i - p_j) + M(1 - \zeta_{i,j}) \geq 0 \tag{3.35}$$

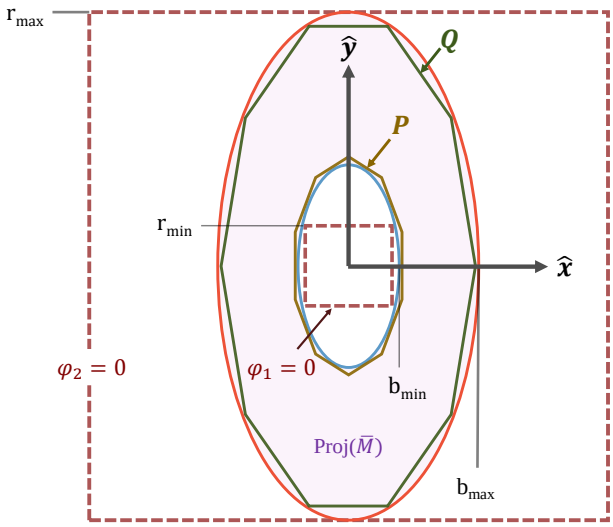


Figure 3.3: Notional depiction of the position constraints defined by the minimum allowable distance $r_{\min}$ and maximum allowable distance $r_{\max}$.

$$\sum_{j=1,\ldots,N_{\mathrm{Q}}} \zeta_{i,j} \geq 1 \tag{3.36}$$

with $i = \tau$ and $M \in \mathbb{R}_+$ (defined sufficiently large as described in Section 1.4) ensure that $\vartheta_3(x_\tau) \geq b_{\min}^2$. Specifically, (3.35) constrains $\zeta_{i,j} = 0$ when the position is on the exterior of the j^{th} edge of $\mathcal{P}$. The requirement that $r_i \notin \mathcal{P}$ is then represented by (3.36). The constraints developed in this section restrict x_τ to a polytope contained in $\bar{\mathcal{M}}$; consequently, (3.32)-(3.36) $\implies$ (3.31). A notional illustration of the polygons $\mathcal{P}$, $\mathcal{Q}$ is shown along with the constraint boundaries in Figure 3.3.

Example Objective Functions

In practice, the appropriate choice of an objective function depends on the specific needs of the mission. The proposed methodology does not assume a particular form for the objective. However, since vehicle efficiency is commonly of critical importance to real-world missions, we find it useful to review here some common approximations for penalizing actuation using quadratic and linear

functions. A simple option is to use the power limiting cost function [119, 120],

$$J = \sum_{i=1}^{\tau} (u_{\mathrm{x},\,i}^2 + u_{\mathrm{y},\,i}^2)\Delta t \tag{3.37}$$

which forms a Mixed Integer Quadratic Program (MIQP). With the introduction of additional constraints, is also possible to use a peice-wise affine approximation of the Euclidean norm of commanded translational acceleration [100]. The resulting cost function is linear,

$$\begin{aligned} &J = \textstyle\sum_{i=1}^{\tau} G_i \\ &s.t. \quad \textstyle\bigwedge_{n=1}^{N_J} \;\; G_i \geq u_{\mathrm{x},i} \sin\left(\frac{2\pi n}{N_\mathrm{J}}\right) + u_{\mathrm{y},i} \cos\left(\frac{2\pi n}{N_\mathrm{J}}\right). \end{aligned} \tag{3.38}$$

The constraints here approximate the second order cone constraints $G_i \geq \|u_i\|_2,\ i = 1, \ldots, \tau$ with an N_J sided polygon. Finally, the cost function $J = \sum_{i=1}^{\tau} \|u_i\|_p$ with $p = 1$ or $p = \infty$ may be represented with a piece-wise linear cost function; see Chapter 9 of [4].

Mixed Integer Program

Given the initial state $x(0) = x_{\mathrm{init}}$, the states and control inputs of a safe reference trajectory leading to the safe parking orbits set $\bar{\mathcal{M}}$ are given by the solution to the optimal control problem,

MIP to Safe Parking Orbit Set $\bar{\mathcal{M}}$

$$
\begin{aligned}
\min_{z} \quad & J(z) \\
\text{s.t.} \quad & x_{i+1} = A_{\text{dt}} x_i + B_{\text{dt}} u_i, && i = 1, ..., \tau - 1 \\
& u_{x,i},\, u_{x,i} \in [-u_{\max},\, u_{\max}], && i = 1, ..., \tau - 1 \\
& x_i = x_{\text{init}}, && i = 1 \\
& (3.24) - (3.30), && i = 2, ..., \tau - 1 \\
& (3.32) - (3.36) && i = \tau
\end{aligned}
\tag{3.39}
$$

where $A_{\text{dt}} = e^{A\Delta t}$ is the state transition matrix, $B_{\text{dt}} = \int_0^{\Delta t} e^{A(t-\tau)} B \mathrm{d}\tau$, the decision vector z contains the states x_i (with $i = 1, ..., \tau$) and control inputs u_i (with $i = 1, ..., \tau - 1$), in addition to the supplemental binary and continuous variables defined in the constraints. Note that the MIP finds both the most efficient parking orbit within the specified range, and the optimal trajectory to that orbit. This program may be modified for the docking problem by removing the collision avoidance constraints, and replacing the endpoint constraint with $x_\tau = 0$.

3.5.1 Regulatory Controllers

Regulation to Planned Trajectory

The MIP developed in the previous section returns a set of $\tau - 1$ control inputs, along with the τ corresponding trajectory states. The ideal state at time t is tracked between updates using a Linear Quadratic Regulator (LQR) as the ancillary control law. The net control at t is,

$$u(t) = u^*(t) + K_{\text{lqr}}(x^*(t) - x(t)) \tag{3.40}$$

where $K_{\text{lqr}} \in \mathbb{R}^{2\times 4}$ is the LQR gain matrix, $u^* \in \mathbb{R}^2$, $x^* \in \mathbb{R}^4$ are the ideal control and state at time t, taken from a linear interpolation of the control and state solutions returned from the MIP.

Experiments use state cost matrix $Q = 0.001\, I_4$ and control cost matrix $R = 1000\, I_2$ to generate K_{lqr}.

Stabilization to Parking Orbit Set

The MIP and the regulatory controller developed in the above sections serve the purpose of safely bringing the spacecraft to the safe parking orbit set $\bar{\mathcal{M}}$. Once the chaser spacecraft has successfully reached this set, the spacecraft switches to a simpler control law that stabilizes the system to the eNMT set $\mathcal{M}$. Let, $e_1 := \vartheta_1$, $e_2 := \vartheta_2$ and note that $e := [e_1,\, e_2]^T = 0$ when $x \in \mathcal{M}$. The *error dynamics* are then,

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1.5n \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} 1/m_c & 0 \\ 0 & 1/m_c \end{bmatrix} \begin{bmatrix} u_{\mathrm{x}} \\ u_{\mathrm{y}} \end{bmatrix}. \tag{3.41}$$

It can be shown that the control law,

$$u_e(e) = \begin{bmatrix} -K & -1.5n \\ 0 & -K \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{3.42}$$

globally stabilizes (3.41) to the origin for $K > 0$; equivalently, it globally stabilizes (3.3) is to the set $\mathcal{M}$. The parameter K is a tunable gain parameter that affects the strength of the response. Experiments in this chapter use $K = 0.6$. Note that since (3.42) drives the system to $\mathcal{M}$, and not the safe subset $\bar{\mathcal{M}}$, it may be necessary in practice to temporarily switch back to the MIP controller if sufficient drift occurs.

3.6 Simulated Case Studies

This section considers case studies for a system with $m_{\mathrm{c}} = 50$ kg, $u_{\max} = 0.5$ N, $n = 0.001027$ rad/s. In all cases, the safety constraints are defined with, $r_{\min} = 0.5$ km, $r_{\max} = 8$ km, $\kappa_1 = 0.5$ m/s, $\kappa_2 = 2n$ s^{-1}, $N_{\mathrm{P}} = N_{\mathrm{Q}} = 12$. In the first example, the performance of the planning

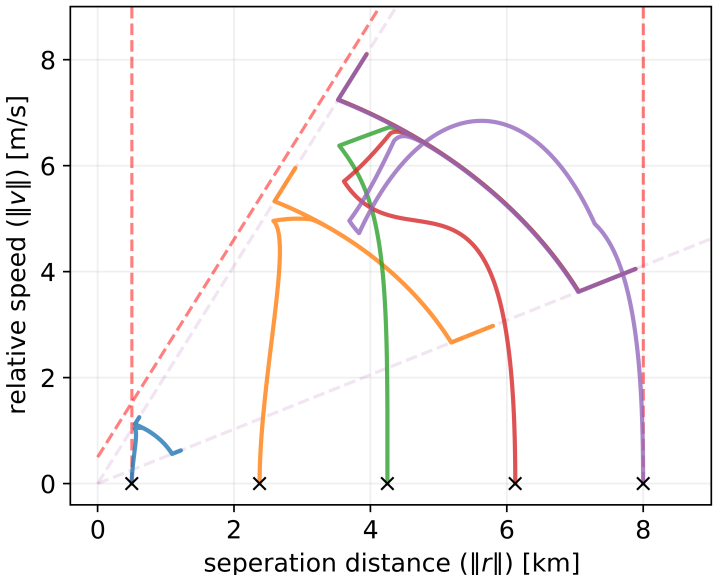


Figure 3.4: Projected trajectories for the deterministic system from initial states with $r_{\mathrm{y}} = v_{\mathrm{x}} = v_{\mathrm{y}} = 0$, and $r_x \in \{500 + 1875i\}_{i=0,\dots,4}\,\mathrm{m}$ using the trajectory planning formulation (3.39) with $T = 1000\,\mathrm{s}$ and $\tau = 101$, and stabilizing controller (3.42).

Table 3.1: Cost Values

x-Position [m]	Ideal Cost [$\mathrm{N^2s}$]	Full Cost [$\mathrm{N^2s}$]
501.0	0.23	2.15
2375.5	5.16	6.62
4250.0	19.18	20.77
6125.5	55.79	56.05
7999.0	116.94	117.53

and control algorithms developed in Section 3.5 are tested under simulated noise and estimation uncertainty. In the second example, the MIP is used to construct an RTA mechanism that filters the input of an unsafe primary controller in such a way that preserves safety of the system.

3.6.1 Cost Comparison

This section compares the performance of the planning formulation developed in Section 3.5 under simulated noise. Ideal trajectories are generated with the MIP (3.39) using cost function (5.31), horizon $T = 1000$ s, and $\tau = 101$ steps. LQR regulation to the ideal states is considered, as described in Section 3.5.1. Though it is not included in the comparison here, one may also consider

receding horizon formulations. The stabilizing controller from Eq. (3.42) takes over at the end of the planning horizon — *i.e.* once $t = T$ and the chaser spacecraft has reached the parking orbit set $\bar{\mathcal{M}}$.

Simulations are conducted from initial states with $r_{\mathrm{y}} = v_{\mathrm{x}} = v_{\mathrm{y}} = 0$, and $r_{\mathrm{x}} \in \{501 + 1874.5i\}_{i=0,\ldots,4}$ m. The controller is sampled at 0.1 Hz, and the system is simulated for 3500 s. The simulated trajectories for the ideal case are shown in Figure 3.4. The simulated cost for the cases with and without uncertainty are indicated in Table 3.1. Here, the *ideal cost* is taken from the reference control points from the MIP, and the *full cost* case is the total cost with the noise and estimation model included, and the regulatory controllers being used to mitigate the resulting state error. The costs for the case including noise effects represents an average over 10 runs. The values indicate that the control system performs well under effects of noise and uncertainty. Note that for the closest case —where a relatively low thrust is commanded over a long time period— most of the actuation effort results from the task of rejecting noise.

3.6.2 Run Time Assurance with MIP-based Backup Controller

We now illustrate the use of the backup controller in the SBSF algorithm (see Section 2.3.3). Here the trajectory planner Eq. (3.39) is used to override the desired actions of a primary controller when necessary for preserving safety. A *nominal* control law $u_{\mathrm{nom}}(x) = Kx$ attempts to drive the chaser spacecraft from the stationary point at $x = [0, -2000\,\mathrm{m}, 0, 0]$ to the stationary point at $x = [0, 750\,\mathrm{m}, 0, 0]$. The matrix K is an LQR gain matrix generated with state cost matrix $Q = 0.0005\, I_4$ and control cost matrix $R = 10000\, I_2$. The trajectory under this control law is shown as the grey line in Figure 3.5. It is apparent from these plots that a direct application of u_{nom} results in a violation of the collision avoidance and relative speed constraints (Eq. (3.7), Eq. (3.9) respectively).

An RTA filter is constructed such that at any state x, the applied control is either the desired control input $u_{\mathrm{nom}}(x)$, or the *backup* control input $u_{\mathrm{b}}(x)$, where u_{b} is given by the solution to Eq. (3.39). Whether or not the desired input is accepted at state x_1 is based on whether a safe

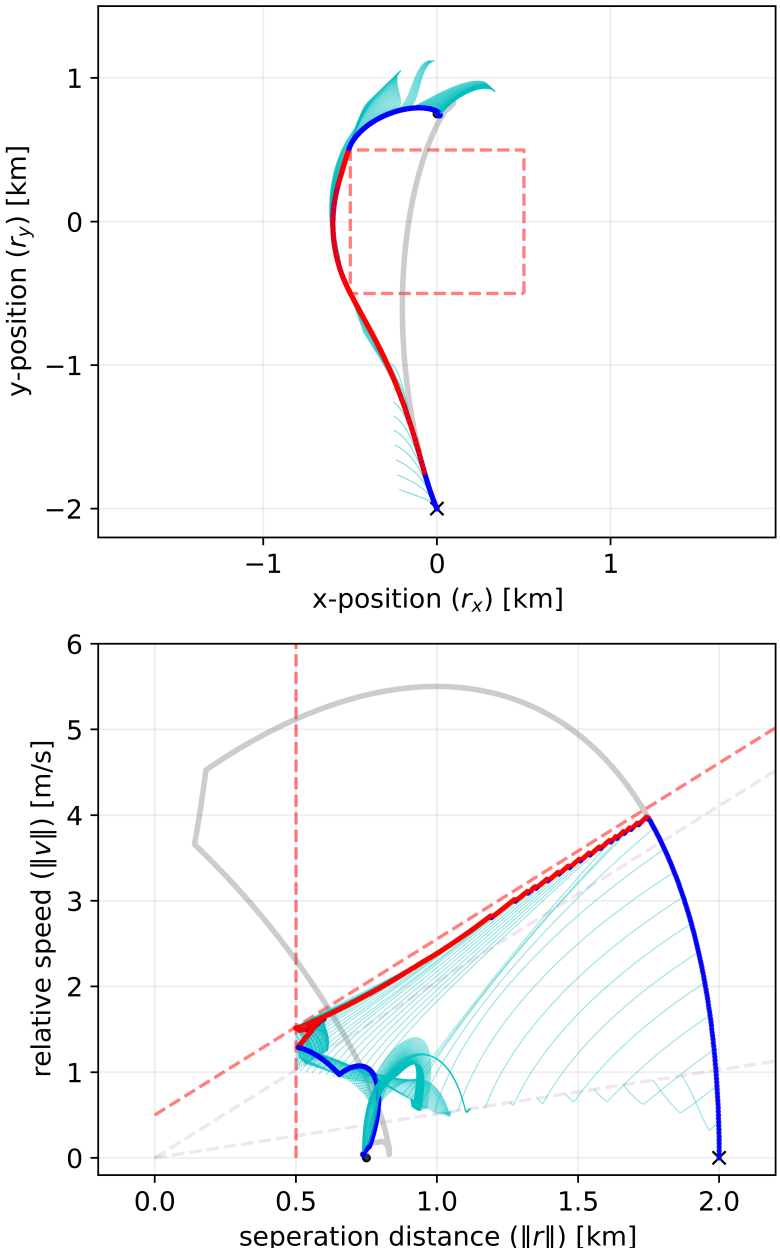


Figure 3.5: Projected trajectories from $x = [0, -2\,\mathrm{km}, 0, 0]^T$ to $x = [0, 0.75\,\mathrm{km}, 0, 0]^T$ over a 2400 s period. The trajectories without RTA are grey. The trajectories with RTA are blue on iterations where the primary controller u_{nom} provides the input, and red when u_{b} provides the input. The backup trajectories (cyan) are shown at 8 s intervals.

trajectory can be found starting with this input. Specifically, let $\tilde{x}_2$ be the estimated state after applying $u_{\text{nom}}(x_1)$ for Δt seconds, and consider the MIP solved with initial condition $\tilde{x}_2$. The backup control $u_{\text{b}}(x_1)$ is commanded if (i) the MIP is infeasible, or (ii) if any of the points on the *backup* trajectory (*i.e.* the solution states) come within 0.05 m/s of the constraint defined by (3.9), or 5 m of the constraints defined by (3.7), (3.8). The first condition ensures that the chaser will only move to new states if it can find a safe solution from that state. The second condition makes the formulation more robust to noise and uncertainty by adding a buffer region on the constraints. Note that if the chaser spacecraft is perturbed into a region where the MIP cannot find a safe solution, then backup control inputs u_{b} may be provided by using (3.40) to regulate to the most recent safe trajectory found.

The RTA is tested with the MIP using a $T = 50$ s horizon composed of $\tau = 50$ points. As illustrated in Figure 3.5, the RTA is effective at preventing violations in the safety constraints. Note that while the RTA enforces constraints, the task of successfully reaching the goal location is allocated to the primary controller. That is, the RTA does not guarantee that the goal will be reached.

CHAPTER 4

RUN TIME ASSURED SPACECRAFT ATTITUDE CONTROL UNDER NONDETERMINISTIC ASSUMPTIONS

This chapter presents a summary of the development and testing of a Run Time Assurance (RTA) filter for a torque-controlled spacecraft in free rotational motion, subject to line-of-sight constraints. A nondeterministic dynamical model is considered and an RTA filter is constructed using recent results from reachability theory in addition to optimization-based computation of invariant sets. Safety guarantees exist when a disturbance torque on the spacecraft is bounded within a defined range. The practicality of the approach is demonstrated with an application on a hardware testbed.

4.1 Preliminaries

This section considers the dynamics

$$\dot{x} = F(x, u, w) \tag{4.1}$$

with state $x \in X \subseteq \mathbb{R}^n$, control input $u \in U \subseteq \mathbb{R}^m$, and disturbance input $w \in W \subset \mathbb{R}^m$. The sets X, U and W denote the state, control and disturbance spaces of (4.1), respectively, and we assume throughout that the disturbance space $W := [\underline{w}, \overline{w}]$ is a hyperrectangle defined by the endpoints $\underline{w}, \overline{w} \in \mathbb{R}^p$.

We denote by $\Phi(t; x, u, w)$ the (assumed unique) state of (4.1) reached at time t when beginning at state $x \in X$ at time 0 and evolving subject to the control input policy $u(t; x) \in U$ and the disturbance signal $w(t) \in W$. Similarly, given a control policy $u(t; x)$,

$$\mathcal{R}(t; A, u) := \{\Phi(t; x, u, w) \mid x \in A, w : [0, t) \to W\} \tag{4.2}$$

denotes the time-t reachable set of (4.1) from the initial set $A \subset X$ under u. In the special instance where (4.1) does not depend on u, so that

$$\dot{x} = F(x, w), \tag{4.3}$$

we omit the third inputs of Φ and $\mathcal{R}$ so that $\Phi(t; x, w)$ denotes the transition map of (4.3) and $\mathcal{R}(t; A)$ denotes the time-t reachable set. Further, when (4.1) does not depend on w the system is said to be *deterministic*; otherwise the system is said to be *nondeterministic*.

4.1.1 Review of Safety for Nondeterministic Dynamical Systems

The system (4.1) is paired with a safety constraint defined on the state $\varphi(x) \geq 0$ and a *constraint set* $\mathcal{C}_{\mathrm{A}} \subset X$ consisting of all states satisfying this constraint, *i.e.* $\mathcal{C}_{\mathrm{A}} := \{x \in X \mid \varphi(x) \geq 0\}$. Feedback controller safety is formalised by a set invariance requirement on the closed-loop dynamics, as discussed next.

Definition 4.1.1 (robust forward invariance)**.** Given a system $\dot{x} = F(x, w)$, as in (4.3), the set $A \subseteq X$ is *robustly forward invariant* for (4.3) if $\Phi(t; x, w) \in A$ for all $x \in A$, all $t \geq 0$ and all piecewise continuous disturbance inputs $w : [0, \infty) \to W$. This is extended to controlled dynamical systems as well so that A is robustly forward invariant for (4.1) under $u(t; x)$ when $\Phi(t; x, u, w) \in A$ for all $t \geq 0$. △

Definition 4.1.2 (robust safety)**.** A set of states $\mathcal{C}_{\mathrm{S}} \subset X$ is said to be *safe* with respect to a nondeterministic dynamical system, a control law, and a constraint set $\mathcal{C}_{\mathrm{A}}$, if that set is a robustly forward invariant subset of the allowable set. That is, if $\mathcal{C}_{\mathrm{S}} \subseteq \mathcal{C}_{\mathrm{A}}$ and $x(t_0) \in \mathcal{C}_{\mathrm{S}} \implies \forall\ \geq t_0,\ x(t) \in \mathcal{C}_{\mathrm{S}}$ for all realizations of the nondeterminism. Furthermore, if $\mathcal{C}_{\mathrm{S}}$ is a safe set, then any state in $\mathcal{C}_{\mathrm{S}}$ is said to be a safe state. △

4.1.2 Efficient Reachability Analysis via Mixed Monotonicity

Reachablity analysis for dynamical systems consists of identifying a set of future system states given a set of initial system states and certain metrics on the class of disturbances in consideration. Modern advances in reachability have produced computationally efficient techniques for approximating reachable sets, with computational speeds suitable for *e.g.* computing reachable sets in the control-loop and enforcing safe behavior online from these predictions [121, 122, 123].

Of particular interest to this work, the *mixed monotonicity* property of dynamical systems has been applied for the efficient overapproximation of reachable sets using hyperrectangles as described below. A dynamical system, possibly subject to a disturbance input, is mixed monotone when there exists a related decomposition function that separates the initial system dynamics into *cooperative* and *competitive* state interactions [124]. Mixed monotonicity applies to continuous-time systems [125, 126, 127, 128, 129], discrete-time systems [130, 131], controlled systems [132, 124], and systems with disturbances [133, 132, 131]; however, for ease of exposition, we consider mainly in this section the uncontrolled continuous-time dynamics (4.3).

Definition 4.1.3 (Mixed Monotonicity [124])**.** Given a locally Lipschitz continuous function $d : X \times X \to \mathbb{R}^n$, the system (4.3) is *mixed monotone with respect to* d if all of the following hold:

1. For all $x \in X$ and all w
$$d(x, w, x, w) = F(x, w).$$

2. For all $i, j \in \{1, \cdots, n\}$, with $i \neq j$,
$$\frac{\partial d_i}{\partial x_j}(x, w, \widehat{x}, \widehat{w}) \geq 0$$
for all $x, \widehat{x} \in X$, and all $w, \widehat{w}$ such that $\frac{\partial d}{\partial x}$ exists.

3. For all $i, j \in \{1, \cdots, n\}$,
$$\frac{\partial d_i}{\partial \widehat{x}_j}(x, w, \widehat{x}, \widehat{w}) \leq 0$$

for all $x, \widehat{x} \in X$, and all $w, \widehat{w}$ such that $\frac{\partial d}{\partial \widehat{x}}$ exists.

4. For all $i \in \{1, \cdots, n\}$ and all $j \in \{1, \cdots, p\}$,

$$\frac{\partial d_i}{\partial w_j}(x, w, \widehat{x}, \widehat{w}) \geq 0 \geq \frac{\partial d_i}{\partial \widehat{w}_j}(x, w, \widehat{x}, \widehat{w})$$

for all $x, \widehat{x} \in X$, and all $w, \widehat{w}$ such that $\frac{\partial d}{\partial w}$ and $\frac{\partial d}{\partial \widehat{w}}$ exist. $\triangle$

When (4.1) is mixed monotone with respect to d, d is a *decomposition function* for (4.1) and

$$\begin{bmatrix} \dot{x} \\ \dot{\widehat{x}} \end{bmatrix} = E(x, \widehat{x}) := \begin{bmatrix} d(x, \underline{w}, \widehat{x}, \overline{w}) \\ d(\widehat{x}, \overline{w}, x, \underline{w}) \end{bmatrix} \tag{4.4}$$

is the *embedding system relative to* d. An important feature of mixed monotone systems that we exploit in this chapter is that overapproximations of reachable sets can be efficiently computed via single simulation of the embedding system. It is convenient to define the following notation: given

$$\Phi = \begin{bmatrix} x \\ \widehat{x} \end{bmatrix}, \tag{4.5}$$

with $x, \widehat{x} \in \mathbb{R}^n$, then the hyperrectangular set formed by the first and last n components of Φ is denoted by

$$[\![\Phi]\!] := [x, \widehat{x}]. \tag{4.6}$$

Additionally, we define the following notation for composing vectors:

$$(x, \widehat{x}) := \begin{bmatrix} x \\ \widehat{x} \end{bmatrix}. \tag{4.7}$$

Proposition 4.1.1. Let (4.3) be mixed monotone with respect to d and choose $A := [\underline{x}, \overline{x}] \subset X$. If $\Phi^E(t; (\underline{x}, \overline{x})) \in X \times X$ for all $t \in [0, T]$ then

$$\mathcal{R}(T; A) \subseteq [\![\Phi^E(T; (\underline{x}, \overline{x}))]\!], \tag{4.8}$$

where $\Phi^E(t; a)$ is the state transition function of the embedding system (4.4). $\Diamond$

The proof of Proposition 4.1.1 appears in [124] and also in [132] and [134].

Proposition 4.1.1 provides an efficient procedure for computing reachable sets for (4.1) via a single simulation of the deterministic embedding system (4.4). The main challenge in this approach, however, is in identifying a suitable decomposition function for (4.3); generally, a mixed monotone system will be mixed monotone with respect to many decomposition functions, however, certain decomposition functions may provide more/less conservative approximations of reachable sets than others when used with Proposition 4.1.1. This is the main point of study in [135] where the authors show that all systems of the form (4.3), bearing a locally Lipshitz continuous vector field, are mixed monotone with respect to a unique *tight decomposition function* that provides a tighter approximation of reachable sets than any other decomposition function for (4.3).

Proposition 4.1.2. [135] Any system of the form (4.3) is mixed monotone with respect to d constructed elementwise according to

$$d_i(x, w, \widehat{x}, \widehat{w}) = \begin{cases} \min\limits_{\substack{y \in [x, \widehat{x}] \\ y_i = x_i \\ z \in [w, \widehat{w}]}} F_i(y, z) & \text{if } (x, w) \preceq (\widehat{x}, \widehat{x}), \\ \\ \max\limits_{\substack{y \in [\widehat{x}, x] \\ y_i = x_i \\ z \in [\widehat{w}, w]}} F_i(y, z) & \text{if } (\widehat{x}, \widehat{w}) \preceq (x, w). \end{cases} \tag{4.9}$$

Moreover, for all other decomposition functions d' for (4.3) and any initial set $A = [\underline{x}, \overline{x}]$,

$$\mathcal{R}(T; A) \subseteq [\![\Phi^E(t; (\underline{x}, \overline{x}))]\!] \subseteq [\![\Phi^{E'}(t; (\underline{x}, \overline{x}))]\!] \tag{4.10}$$

where Φ^E and $\Phi^{E'}$ denote the state transition functions of the embedding systems constructed from d and d', respectively. $\Diamond$

As posed in (4.9), computing a tight decomposition function requires solving a generally non-convex optimization problem for each quadruple $(x, w, \widehat{x}, \widehat{w})$. However, in certain instances it is possible to compute a tight decomposition function in closed form; see [135] for representative examples. Regardless, the general computational infeasibility of (4.9) implies that it is of limited direct use computationally and, in practice, decomposition functions are generally not obtained using Proposition 4.1.2. For this reason, decomposition functions constructed from other means can be preferable; see [132, 136] for an algorithm for computing decomposition functions for systems with uniformly bounded Jacobian matrices and see [134] for an algorithm for computing decomposition functions for systems defined by polynomial vector fields.

4.1.3 Quaternions

Consider $i,\ j,\ k$ be defined such that,

$$i^2 + j^2 + k^2 = ijk = -1 \tag{4.11}$$

A quaternion $q \in \mathbb{H} \subset \mathbb{R}^4$ consists of four parameters $q = [q_0, q_1, q_2, q_3]^T$ such that

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{4.12}$$

and these parameters may be used to represent rotations, *e.g.*

$$q = \cos(\alpha/2) + \sin(\alpha/2)(u_1 i + u_2 j + u_3 k) \tag{4.13}$$

where $u_1,\ u_2,\ u_3$ are the components of a unit vector defining an axis of rotation and α is the angle of rotation about that axis. Given a vector $p = [p_1, p_2, p_3] \in \mathbb{R}^3$, and a vector $r = [r_1, r_2, r_3] \in \mathbb{R}^3$ that is obtained by rotating p, a quaternion exists such that $r = q\, p\, q^{-1}$. Equivalently, the quaternion

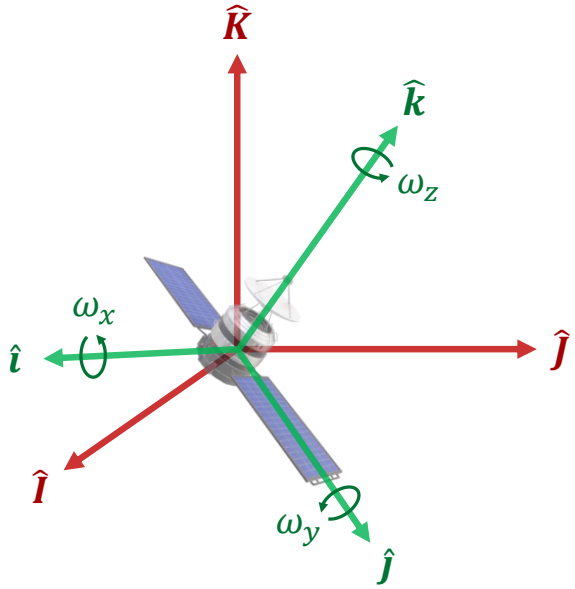


Figure 4.1: Depiction of spacecraft body frame $\mathcal{F}_{\mathrm{B}}$ and inertial frame $\mathcal{F}_{\mathrm{I}}$.

may be used to form a rotation matrix:

$$\mathrm{R}(q) = \begin{bmatrix} (1-2q_2^2-2q_3^2) & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & (1-2q_1^2-2q_3^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & (1-2q_1^2-2q_2^2) \end{bmatrix}. \tag{4.14}$$

such that $r = \mathrm{R}(q)\, p$.

4.2 Safety and Dynamics for Spacecraft Attitude Control

4.2.1 Spacecraft Attitude Dynamics

The system of interest consists of a torque-controlled spacecraft in unconstrained rotational motion. The spacecraft is modeled with state $x = [q^T, \omega^T]^T \in \mathbb{R}^7$ and control input $u \in U \subset \mathbb{R}^3$, where the orientation of the spacecraft is captured by the quaternion vector $q \in \mathbb{H} \subset \mathbb{R}^4$, the angular velocity is captured by $\omega \in \mathbb{R}^3$, and u describes an applied torque input. The quaternion vector $q = [q_0, q_1, q_2, q_3]^T$ defines the orientation of a body-fixed reference frame $\mathcal{F}_{\mathrm{B}}$ with respect to an inertial frame $\mathcal{F}_{\mathrm{I}}$, and the components of $\omega = [\omega_1, \omega_2, \omega_3]^T$ and $u = [u_1, u_2, u_3]^T$ are expressed in

the body frame $\mathcal{F}_{\mathrm{B}}$. A depiction of the axes is shown in Figure 4.1. We let $\mathcal{F}_{\mathrm{I}} := (\mathcal{O}, \widehat{I}, \widehat{J}, \widehat{K})$ and $\mathcal{F}_{\mathrm{B}} := (\mathcal{O}, \widehat{i}, \widehat{j}, \widehat{k})$ so that, without loss of generality, the origin of the inertial frame $\mathcal{O}$ is the same as that of the body frame and the unit vectors $\widehat{i}$, $\widehat{j}$, $\widehat{k}$ and $\widehat{I}$, $\widehat{J}$, $\widehat{K}$ form right-handed orthogonal bases for $\mathcal{F}_{\mathrm{B}}$ and $\mathcal{F}_{\mathrm{I}}$, respectively, *i.e.* $\widehat{i} \times \widehat{j} = \widehat{k}$ and $\widehat{I} \times \widehat{J} = \widehat{K}$.

The dynamics of the spacecraft are

$$\begin{aligned} \dot{q} &= Q(q)\omega \\ \dot{\omega} &= J^{-1}(-\omega \times J\omega + u + w) \end{aligned} \tag{4.15}$$

where the bounded, nondeterministic input variable $w \in W \subset \mathbb{R}^3$ models external torque perturbations, $Q(q)$ is the quaternion kinematic matrix, given by

$$Q(q) = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{4.16}$$

and $J \in \mathbb{R}^{3\times 3}$ is the inertia matrix of the spacecraft taken with respect to $\mathcal{F}_{\mathrm{B}}$. The inertia matrix is *not* assumed to be diagonal.

4.2.2 Line-of-Sight Constraint

Safety of the system is defined by a line-of-sight (LOS) constraint that restricts the attitude of the vehicle so that a *boresight* direction lies within some maximum angle of a defined safe direction. Let $\widehat{b} \in \mathbb{R}^3$ be a body-fixed unit vector aligned with the boresight direction, and $\widehat{c} \in \mathbb{R}^3$ be an inertially-fixed unit vector defining a *safe* pointing direction for $\widehat{b}$. The LOS angle $\beta \in [0, \pi]$ rad is defined as the angle subtended between these two vectors and the safety constraint is

$$\beta \leq \beta_{\max}, \tag{4.17}$$

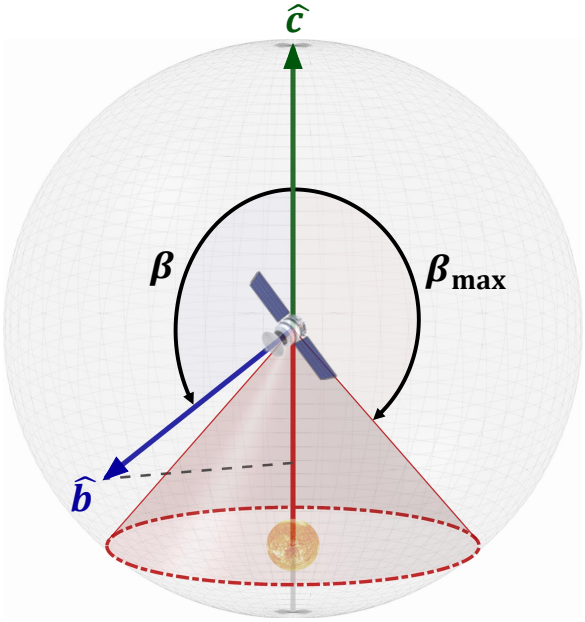


Figure 4.2: Depiction of parameters defining safe configuration. $\widehat{b}$ is the boresight vector, $\widehat{c}$ is the safe direction, and β is the line-of-sight angle. Note that $\beta > \frac{\pi}{2}$ so that the system is constrained to remain outside of an avoid cone (red) centered on $-\widehat{c}$.

where $\beta_{\max} \in [0, \pi]$ rad. If $\widehat{b}$ is expressed in the coordinates of a body-fixed frame $\mathcal{F}_{\mathrm{B}}$ and $\widehat{c}$ is expressed in the coordinates of an inertial frame $\mathcal{F}_{\mathrm{I}}$, then

$$\cos(\beta) = \widehat{c}^{\,T} \mathrm{R}(q)\, \widehat{b} \tag{4.18}$$

where $q \in \mathbb{H} \subset \mathbb{R}^4$ is a quaternion defining the rotation from $\mathcal{F}_{\mathrm{I}}$ to $\mathcal{F}_{\mathrm{B}}$, and $\mathrm{R} : \mathbb{H} \to SO(3)$ is given by (4.14). Note that $\cos(\beta)$ represents the projection of the boresight direction onto the safe axis. In the case where $\beta < \frac{\pi}{2}$, the boresight direction is restricted to a cone centered around $\widehat{c}$ and in the case where $\beta > \frac{\pi}{2}$, it is constrained to lie outside of a cone centered around $-\widehat{c}$. The geometry is depicted in Figure 4.2.

The safety constraint can be greatly simplified with a convenient choice of frames. In this chapter we assume, without loss of generality, that $\mathcal{F}_{\mathrm{B}}$ is defined such that $\widehat{b} = \widehat{k} = [0, 0, 1]^T$ and

$\mathcal{F}_{\mathrm{I}}$ is defined such that $\widehat{c} = \widehat{K} = [0, 0, 1]^T$. In this case, it is clear that

$$\cos(\beta) = 1 - 2q_1^2 - 2q_2^2, \tag{4.19}$$

and the safety constraint (4.17) is equivalently represented as

$$\varphi(x) := 1 - \cos(\beta_{\max}) - 2q_1^2 - 2q_2^2 \geq 0. \tag{4.20}$$

The set of all states that satisfy this constraint is referred to as the *constraint set* and is denoted by

$$\mathcal{C}_{\mathrm{A}} := \{x \in X \,|\, \varphi(x) \geq 0\} \tag{4.21}$$

and the boundary of this set is

$$\partial\mathcal{C}_{\mathrm{A}} := \{x \in \mathcal{C}_{\mathrm{A}} \,|\, \varphi(x) = 0\}. \tag{4.22}$$

As discussed previously, there will generally exist states in $\mathcal{C}_{\mathrm{A}}$ for which a departure is inevitable. A safe set of states consists of a subset of the constraint set that is robustly forward invariant under a given control law.

4.2.3 Backup Controller

In this section a backup controller (or recovery controller) $u_{\mathrm{b}} : X \to U$ is constructed to drive the system to an invariant subset of the constraint set $\mathcal{C}_{\mathrm{A}}$. Noting that $\varphi(x)$ is maximized when $\beta = 0$ and that invariant points occur whenever $\omega = 0$, a controller is introduced that stabilizes the system to this configuration. The backup controller for the system is

$$u_{\mathrm{b}}(x) = \sigma(\omega \times J\omega - k_{\mathrm{p}} J\eta - k_{\mathrm{d}} J\omega), \tag{4.23}$$

where $k_{\mathrm{p}} > 0$, $k_{\mathrm{d}} > 0$ are controller gains, where

$$\eta = \begin{bmatrix} 2(q_2 q_3 + q_0 q_1) \\ 2(q_0 q_2 - q_1 q_3) \\ 0 \end{bmatrix}, \tag{4.24}$$

and where

$$\sigma(u) = \begin{bmatrix} u_{\max} \tanh(\frac{u_1}{u_{\max}}) \\ u_{\max} \tanh(\frac{u_2}{u_{\max}}) \\ u_{\max} \tanh(\frac{u_3}{u_{\max}}) \end{bmatrix}. \tag{4.25}$$

The purpose of the saturation function $\sigma : \mathbb{R}^m \to U$ is to bound all inputs in $\mathbb{R}^m$ to the admissible input set $U = [-u_{\max}, u_{\max}]^3$, while approximating the input within this domain; *i.e.* $\sigma(u) \in U$ for all $u \in \mathbb{R}^m$ and $u \approx \sigma(u)$ when $u \in U$.

4.3 Mixed Monotonicity and Reachability Analysis for Systems with Outputs

Before presenting the main result in Section 4.4, we first present several theoretical results related to the mixed monotonicity of spacecraft systems. We begin by showing that the system (4.15) is mixed monotone, and that the quaternion dynamics are mixed monotone with a tight decomposition function. Next, we explore conservatism in the approximation of reachable sets for (4.15) and we present one technique for reducing conservatism in the application of Proposition 4.1.1 based on the results of [137].

4.3.1 Decomposing the Spacecraft Dynamics

The RTA architecture, presented later in Section 4.4, requires decomposition functions for two dynamical systems: the *open-loop dynamics*

$$\dot{q} = Q(q)\omega \tag{4.26}$$

$$\dot{\omega} = -J^{-1}(\omega \times J\omega) + J^{-1}u + J^{-1}w, \tag{4.27}$$

presented previously as system (4.15), where now $u \in U$ is assumed constant, and the *closed-loop backup dynamics*

$$\dot{q} = Q(q)\omega \tag{4.28}$$

$$\dot{\omega} = -J^{-1}(\omega \times J\omega) + J^{-1}\sigma(\omega \times J\omega - k_p J\eta - k_d J\omega) + J^{-1}w \tag{4.29}$$

that come from applying (4.23) to (4.15). As such, the first result is to show that both the open-loop dynamics (4.26)–(4.27) and the backup dynamics (4.28)–(4.29) are mixed monotone with decomposition functions given in closed form.

Proposition 4.3.1. The open-loop dynamics (4.26)–(4.27) are mixed monotone. ◊

Proposition 4.3.2. The backup dynamics (4.28)–(4.29) are mixed monotone. ◊

We construct decomposition functions for (4.26)–(4.27) and (4.28)–(4.29) using an iterative process, whereby individual decompositions are constructed for each monomial of *polynomial* dynamics. Using this approach, we find that the quaternion dynamics $\dot{q} = Q(q)\omega$, which appears as both (4.27)/(4.29), is mixed monotone with a tight decomposition function when the term ω is viewed as a disturbance input. Access to a tight decomposition function (even for solely the quaternion dynamics) provides relative confidence that performance cannot be improved substantially when other decomposition functions are used with Proposition 4.1.1 and, since $\varphi(x)$ is dependant only on q_1 and q_2, tight approximations in the quaternion coordinates of (4.15) are paramount.

4.3.2 Reducing Conservatism in Mixed Monotone Reachable Set Approximations

Unlike the *monotonicity* property of dynamical systems, which can be applied to compute *e.g.* the tightest hyperrectangle containing a reachable set [125], the application of the mixed monotonicity property is known to generally result in conservative reachable set approximations. This is the main point of study in [137] where the authors identify four main kinds of conservatism in the application of Proposition 4.1.1, and we have discussed already how *some* conservatism is mitigated when a tight decomposition function is used in the procedure [135]. A second solution posited in [137] for

reducing conservatism in the application of the mixed monotonicity property involves considering a linear transformation of the dynamics. That is, rather than apply Proposition 4.1.1 directly to $\dot{x} = F(x, w)$, one can instead consider a linear transformation on the initial system statespace $x' = Tx$ for some nonsingular transformation matrix $T \in \mathbb{R}^{n\times n}$ and apply Proposition 4.1.1 to the transformed dynamics

$$\dot{x}' = T^{-1}F(Tx', w). \tag{4.30}$$

A key observation in [137] is that applying Proposition 4.1.1 to (4.30) allows for computing *parallelotope* reachable set approximations for the initial system (4.3) and can be used to reduce conservatism in the approach.

In this section, we revisit the open-loop spacecraft dynamics (4.26)–(4.27) and the closed-loop backup dynamics (4.28)–(4.29). We consider, in particular, a state transformation on the angular velocity

$$\omega' = J\omega. \tag{4.31}$$

so that the open-loop spacecraft dynamics (4.26)–(4.27) become

$$\dot{q} = Q(q)J^{-1}\omega' \tag{4.32}$$

$$\dot{\omega}' = -J^{-1}\omega' \times \omega' + u + w, \tag{4.33}$$

and so that the closed-loop backup dynamics (4.28)–(4.29) become

$$\dot{q} = Q(q)J^{-1}\omega' \tag{4.34}$$

$$\dot{\omega}' = -J^{-1}\omega' \times \omega' + \sigma(J^{-1}\omega' \times \omega' - k_p J\eta - k_d\omega') + w. \tag{4.35}$$

Both (4.32)–(4.33) and (4.34)–(4.35) are mixed monotone, and the state transformation (4.31) is observed to reduce conservatism in the approximation of reachable sets.

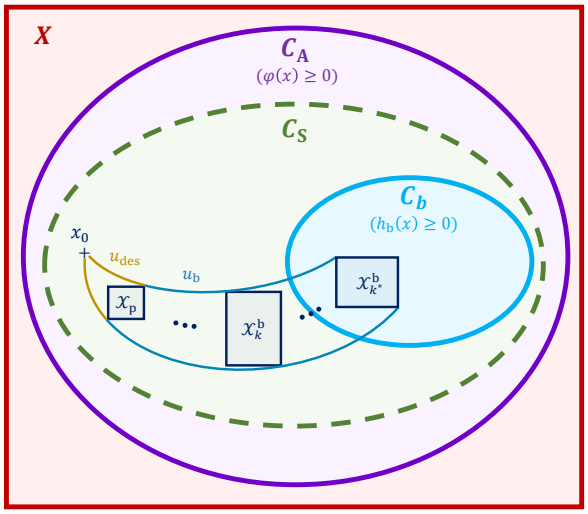


Figure 4.3: Topological depiction of constraint set $\mathcal{C}_\mathrm{A}$, safe terminal set $\mathcal{C}_\mathrm{b}$, safe backwards image $\mathcal{C}_\mathrm{S}$, probe set $\mathcal{X}_\mathrm{p}$, and reachable set overapproximations (RSOs) $\mathcal{X}_k^\mathrm{b}$.

4.4 Run Time Assurance Approach

This section develops the Mixed Monotone Simplex Architecture (MiMoSa), an RTA-based approach for enforcing safety constraints. The algorithm is applied to the spacecraft system (4.15) and the LOS constraint (4.20). The objective is to define a run time assured control law $u_\mathrm{act} : X \times U \to U$ that switches to the recovery controller u_b as necessary to prevent constraint violations. Potential constraint violations are detected by determining whether or not an overapproximation of the reachable set around the recovery trajectory is contained in the constraint set over a finite-time, and that all possible endpoints terminate in a predefined *safe* terminal set. The reachable set overapproximation (RSO) is obtained from a finite-time simulation of the embedding dynamics, as described in Section 4.1.2. Section 4.4.1 describes a safe terminal set contained in the constraint set, and Section 4.4.2 defines a larger safe set from the states that can safely reach the terminal set under the recovery maneuver. Finally, Section 4.4.3 presents the RTA control law and its associated safety guarantees.

4.4.1 Safe Terminal Set

The safe terminal set $\mathcal{C}_{\mathrm{b}}$ provides a sufficient condition for terminating the online RTA look-ahead. Such a set is generally computed offline, either analytically or via optimization techniques, and inserted into the RTA filtering mechanism as an assurance that the backup controller will lead to safety. The set should be robustly invariant, meaning that the set is still invariant even when uncertainties and disturbances in the system model and input signal(s) are present.

A Lyapunov function is used to guide the computation of the invariant set via an optimization procedure. Specifically, the safe terminal set is given by

$$\mathcal{C}_{\mathrm{b}} = \{x \in X \,|\, h_{\mathrm{b}}(x) \geq 0\} \tag{4.36}$$

where $h_{\mathrm{b}}(x) = \gamma - V(x)$ with $V(x)$ being a Lyapunov function for the dynamics under the backup controller on the domain where $\beta < \pi$, and the positive constant $\gamma > 0$ is chosen such that $\mathcal{C}_{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}}$.

4.4.2 Robustly Safe Backward Image of the Terminal Set

While $\mathcal{C}_{\mathrm{b}}$ forms a safe set under u_{b}, the Lyapunov-based approach yields a set that is very conservative, and therefore not practical for use directly in an RTA framework. That is, an RTA that bounds the system trajectories to $\mathcal{C}_{\mathrm{b}}$ would be overly restrictive. However, $\mathcal{C}_{\mathrm{b}}$ can be utilized in a way that grants the system access to a much larger *implicitly defined* safe set via the trajectories of the embedding system. The goal of this section is to define such a set. Specifically, we introduce a set $\mathcal{C}_{\mathrm{S}}$ such that (i) $\mathcal{C}_{\mathrm{S}}$ is robustly forward invariant under u_{b}, and (ii) it is possible to verify whether a particular subset of X is contained in $\mathcal{C}_{\mathrm{S}}$ using the information obtained from integrating the embedding state over a finite-time horizon $[0, T] \subset \mathbb{R}$. For this purpose it is useful to recall that the embedding state provides an overapproximation of the forward reachable set of a system; see Proposition 4.1.1 for details. Given an embedding system, a backup controller, and a safe terminal set under this backup controller, such a set is defined as follows:

Definition 4.4.1 (robustly safe backward image)**.** The *robustly safe backward image* (RSBI) of a set $\mathcal{C}_\mathrm{b}$ is defined with respect to the time horizon $[0, T] \subset \mathbb{R}$, a backup controller u_b, and an embedding system (4.4) as

$$\mathcal{C}_\mathrm{S} := \{x \in X \,|\, \exists t^* \in [0, T] \; s.t. \; \vartheta_1(x, t^*) \wedge \vartheta_2(x, t^*)\} \tag{4.37}$$

where,

$$\begin{aligned} \vartheta_1(x, t^*): &\quad \forall\, t \in [0, t^*],\ [\![\Phi^E_{u_\mathrm{b}}(t; (x, x))]\!] \subseteq \mathcal{C}_\mathrm{A} \\ \vartheta_2(x, t^*): &\quad [\![\Phi^E_{u_\mathrm{b}}(t^*; (x, x))]\!] \subseteq \mathcal{C}_\mathrm{b} \end{aligned}$$

with $\mathcal{C}_\mathrm{b}$ is defined as in Section 4.4.1, and $\Phi^E_{u_\mathrm{b}}(t; a)$ is the state transition function of the embedding system derived from the closed-loop dynamics (4.15) under the controller u_b. $\triangle$

The endpoint constraint $\vartheta_2(x, t^*)$ encodes the condition that the reachable set overapproximation (RSO) is contained in $\mathcal{C}_\mathrm{b}$ by some time $t^* \in [0, T]$, and the intermediate point constraint $\vartheta_1(x, t^*)$ encodes that the RSO is contained in $\mathcal{C}_\mathrm{A}$ until that time. As stated in the following proposition, the RSBI defines a safe set.

Proposition 4.4.1 (Safety of the RSBI under u_b)**.** Consider the RBSI $\mathcal{C}_\mathrm{S}$ defined by (4.37) with respect to the terminal set $\mathcal{C}_\mathrm{b}$ described in Section 4.4.2, the constraint set $\mathcal{C}_\mathrm{A}$ defined by (4.20), the embedding system (4.4), and the control law u_b given by (4.23). Then, $\mathcal{C}_\mathrm{S}$ is robustly forward invariant under the closed-loop dynamics of (4.15) with respect to u_b. Furthermore, $\mathcal{C}_\mathrm{S} \subseteq \mathcal{C}_\mathrm{A}$. $\Diamond$

Proof. By the intermediate condition ϑ_1, if $x \in \mathcal{C}_\mathrm{S}$ then $[\![\Phi^E_{u_\mathrm{b}}(t, (x, x))]\!] \subseteq \mathcal{C}_\mathrm{A}$ for all $t \in [0, T]$. Since $\mathcal{R}_{u_\mathrm{b}}(t; x) \subseteq [\![\Phi^E_{u_\mathrm{b}}(t, (x, x))]\!]$, it is clear that $\mathcal{R}_{u_\mathrm{b}}(t_k; x) \subseteq \mathcal{C}_\mathrm{S}$ for all $t \in [0, T]$. By the endpoint condition ϑ_2, $x \in \mathcal{C}_\mathrm{S}$ implies that $\mathcal{R}_{u_\mathrm{b}}(T; x) \subseteq \mathcal{C}_\mathrm{b}$. Since $\mathcal{C}_\mathrm{b}$ is a robustly forward invariant subset of $\mathcal{C}_\mathrm{A}$, it is clear that for all $t \geq T$, $\mathcal{R}_{u_\mathrm{b}}(t_k; x) \subseteq \mathcal{C}_\mathrm{b} \subseteq \mathcal{C}_\mathrm{A}$. $\square$

Intuitively, the proposition guarantees that for any initial condition in $\mathcal{C}_\mathrm{S}$, the closed-loop system under u_b will continue to satisfy the LOS constraint for all $t \geq 0$.

Approximation of the RSBI

The intermediate constraints on the RSBI are defined on the embedding trajectory $\Phi^E_{u_\mathrm{b}}(t;(x,x))$ over the finite-time horizon $[0,T]$. However, since $[0,T]$ is a continuous interval, this amounts to verifying an infinite set of constraints. In practice, it is necessary to *approximate* $\mathcal{C}_\mathrm{S}$ by evaluating the constraints at a discrete set of points along the interval. While other choices are possible, a reasonable set of points for the discretization corresponds to the instants in time at which the controller updates. Given the controller update period Δt, and using zero as a reference time, a K-step finite-time horizon is defined as

$$\mathcal{T}_K := \{t \in \mathbb{R} \mid t = (k-1)\Delta t,\ k = 1, ..., K\} \tag{4.38}$$

where by design: $(K-1)\Delta t = T$. Furthermore, an infinite-horizon is defined as

$$\mathcal{T}_\infty = \{t \in \mathbb{R} \mid t = k\Delta t,\ k \in \mathbb{Z}_{\geq 0}\}. \tag{4.39}$$

Importantly, the embedding state can be obtained at these points by numerically integrating the embedding system dynamics.

Remark 4.4.1. If the constraints are tightened in a way that assures trajectories do not leave between the time instants in $\mathcal{T}_K$ (see for example [76]), then the approximation of $\mathcal{C}_\mathrm{S}$ is robustly forward invariant under u_b. The potential constraint violation between time-steps is assumed to be negligible in this research and, as such, no distinction is made between the RSBI and the approximation described in this section. Both are denoted with $\mathcal{C}_\mathrm{S}$. △

Test for Set Containment

In this section, a sufficient condition is developed for verifying that a rectangular set $\mathcal{X} := [\underline{x}, \overline{x}]$ is contained in $\mathcal{C}_\mathrm{S}$ using a finite-time simulation of the embedding system. The flow of the embedding

system under u_{b} from this initial set is denoted

$$\begin{bmatrix} \underline{x}_k^{\mathrm{b}} \\ \overline{x}_k^{\mathrm{b}} \end{bmatrix} := \Phi_{u_{\mathrm{b}}}^E(t_k; (\underline{x}, \overline{x})) \tag{4.40}$$

where t_k is the k^{th} timestep in the horizon $\mathcal{T}_K$. Additionally,

$$\underline{x}_k^{\mathrm{b}} = [\underline{q}_{0,k}^{\mathrm{b}}, \underline{q}_{1,k}^{\mathrm{b}}, \underline{q}_{2,k}^{\mathrm{b}}, \underline{q}_{3,k}^{\mathrm{b}}, \underline{\omega}_{1,k}^{\mathrm{b}}, \underline{\omega}_{2,k}^{\mathrm{b}}, \underline{\omega}_{3,k}^{\mathrm{b}}]^T \tag{4.41}$$

and

$$\overline{x}_k^{\mathrm{b}} = [\overline{q}_{0,k}^{\mathrm{b}}, \overline{q}_{1,k}^{\mathrm{b}}, \overline{q}_{2,k}^{\mathrm{b}}, \overline{q}_{3,k}^{\mathrm{b}}, \overline{\omega}_{1,k}^{\mathrm{b}}, \overline{\omega}_{2,k}^{\mathrm{b}}, \overline{\omega}_{3,k}^{\mathrm{b}}]^T. \tag{4.42}$$

The RSO of $\mathcal{X}$ that is obtained from the embedding system flow under u_{b} is denoted as,

$$\mathcal{X}_k^{\mathrm{b}} := [\![\Phi_{u_{\mathrm{b}}}^E(t_k; (\underline{x}, \overline{x}))]\!]. \tag{4.43}$$

A test for containment of $\mathcal{X}$ in $\mathcal{C}_{\mathrm{S}}$ is given by[1]:

$$[\exists\, k^* \in \{1, ..., K\}\ s.t.\ \vartheta_3(\mathcal{X}, k^*)\ \wedge\ \vartheta_4(\mathcal{X}, k^*)] \implies [\mathcal{X} \subseteq \mathcal{C}_{\mathrm{S}}] \tag{4.44}$$

where

$$\begin{aligned} \vartheta_3(\mathcal{X}, k^*): &\quad \forall k \in \{1, \ldots, k^*\},\ \mathcal{X}_k^{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}} \\ \vartheta_4(\mathcal{X}, k^*): &\quad \mathcal{X}_{k^*}^{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{b}} \end{aligned} \tag{4.45}$$

This test is valid due to a well known nesting property of the embedding system; see [124] for further details. To evaluate $\vartheta_3(\mathcal{X}, k^*)$, we note that the following are equivalent

$$(i)\ \mathcal{X}_k^{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{A}} \qquad (ii)\ \min_{x \in \mathcal{X}_k^{\mathrm{b}}} \varphi(x) \geq 0 \tag{4.46}$$

[1]In this context, the set $\mathcal{C}_{\mathrm{S}}$ refers to the approximation of the RSBI, which underapproximates the RSBI under appropriate assumptions on the step size and tightening of the constraints, as discussed in Remark 4.4.1.

where $\varphi(x)$ is given by (4.20). It can be shown that the minimum is,

$$\min_{x \in \mathcal{X}_k^{\mathrm{b}}} \varphi(x) = 1 - \cos(\beta_{\max}) - 2\max(\underline{q}_{1,k}^2, \overline{q}_{1,k}^2) - 2\max(\underline{q}_{2,k}^2, \overline{q}_{2,k}^2) \tag{4.47}$$

Similarly, to evaluate $\vartheta_4(\mathcal{X}, k^*)$, it is observed that the following are equivalent

$$(i)\ \mathcal{X}_{k^*}^{\mathrm{b}} \subseteq \mathcal{C}_{\mathrm{b}} \qquad (ii)\ \min_{x \in \mathcal{X}_{k*}^{\mathrm{b}}} h_{\mathrm{b}}(x) \geq 0. \tag{4.48}$$

4.4.3 Algorithm

The presented control law is defined such that the desired input $u_{\mathrm{des}} \in U$ is applied whenever it can be verified that the input will not result in a departure from the RSBI, and it will be replaced with the input from the recovery controller u_{b} otherwise. Specifically, safety is assessed by taking a *probe step* under the desired input, and determining if all states reachable under this step are safe; *i.e.* contained in $\mathcal{C}_{\mathrm{S}}$. Given an initial state $x_0 \in X$, the *probe set* is defined as

$$\mathcal{X}_{\mathrm{p}} := [\![\Phi^E(\Delta t; (x_0, x_0), u_{\mathrm{des}})]\!], \tag{4.49}$$

where $\Phi^E(t; a, u)$ is the state transition function of the embedding system derived from the open-loop dynamics of (4.15) with the applied control input u. Note that this set is an overapproximation of $\mathcal{R}(\Delta t; x_0, u_{\mathrm{des}})$. The run time assured control law $u_{\mathrm{act}} : X \times U \to U$ is

$$u_{\mathrm{act}}(x, u_{\mathrm{des}}) = \begin{cases} u_{\mathrm{des}} & \text{if} \quad \vartheta_{\mathrm{safe}}(\mathcal{X}_{\mathrm{p}}) \\ u_{\mathrm{b}}(x) & \text{otherwise} \end{cases} \tag{4.50}$$

where

$$\vartheta_{\mathrm{safe}}(\mathcal{X}_{\mathrm{p}}) :\ [\exists\, k^* \in \{1, ..., K\}\ s.t.\ \vartheta_3(\mathcal{X}_{\mathrm{p}}, k^*)\ \wedge\ \vartheta_4(\mathcal{X}_{\mathrm{p}}, k^*)]. \tag{4.51}$$

As described in Section 4.4.2, $\vartheta_{\text{safe}}(\mathcal{X}_{\text{p}})$ can be evaluated with the information obtained from a simulation of the embedding system over $\mathcal{T}_K$. The safety guarantee for this controller is stated in the following theorem.

Theorem 4.4.1 (Safety of the RSBI under u_{act})**.** *Consider the initial state $x(0) \in X$, the times $t_k \in \mathcal{T}_\infty$, and the control law u_{act} described by* (4.50) *and defined with respect to the horizon $\mathcal{T}_K$. The system* (4.15) *under u_{act} obeys the following property,*

$$x(0) \in \mathcal{C}_{\text{S}} \quad \Longrightarrow \quad \forall t_k \in \mathcal{T}_\infty, \quad x(t_k) \in \mathcal{C}_{\text{S}} \tag{4.52}$$

$\Diamond$

Proof. Suppose *ad absurdum* that the implication in (4.52) is false. Then, there must exist some $k \in \mathbb{Z}_{>0}$ such that

$$x(t_k) \in \mathcal{C}_{\text{S}} \ \wedge \ x(t_{k+1}) \notin \mathcal{C}_{\text{S}} \tag{4.53}$$

The control input at t_k is $u_{\text{act}}(x(t_k), u_{\text{des}})$ which must take a value in the set $\{u_{\text{des}}, u_{\text{b}}(x(t_k))\}$. Suppose that $u_{\text{act}} = u_{\text{des}}$, then $x(t_{k+1}) \in \mathcal{X}_{\text{p}}$ and since $x(t_{k+1}) \notin \mathcal{C}_{\text{S}}$, we know that $\mathcal{X}_{\text{p}} \not\subseteq \mathcal{C}_{\text{S}}$. However, in order to have chosen $u_{\text{act}} = u_{\text{des}}$, it must be true that $\vartheta_3(\mathcal{X}_{\text{p}}) \wedge \vartheta_4(\mathcal{X}_{\text{p}})$, which implies that $\mathcal{X}_{\text{p}} \subseteq \mathcal{C}_{\text{S}}$. Since this is not possible, it is apparent that u_{act} could not be equal to u_{des}. Now suppose that $u_{\text{act}} = u_{\text{b}}(x(t_k))$, then $\mathcal{R}_{u_{\text{b}}}(\Delta t; x(t_k)) \not\subseteq \mathcal{C}_{\text{S}}$ since $x(t_{k+1}) \in \mathcal{R}_{u_{\text{b}}}(\Delta t; x(t_k))$ and $x(t_{k+1}) \notin \mathcal{C}_{\text{S}}$. However, by Proposition 4.4.1, $\mathcal{R}_{u_{\text{b}}}(\Delta t; x(t_k))$ must be contained in $\mathcal{C}_{\text{S}}$ as $x(t_k) \in \mathcal{C}_{\text{S}}$. Hence u_{act} can not be equal to u_{b}. Since u_{act} cannot take a value causing a departure from $\mathcal{C}_{\text{S}}$, we have arrived at a contradiction. Consequently, it is clear that Theorem 4.4.1 must be true. $\square$

Informally, if the system starts in a safe configuration ($x(0) \in \mathcal{C}_{\text{S}}$), then it will stay safe for all time. In addition, since u_{b} stabilizes the system to $\mathcal{C}_{\text{b}} \subseteq \mathcal{C}_{\text{A}}$, it is apparent that if the system starts in an unsafe configuration, then it will be driven to a safe configuration, assuming the initial state is in the region of attraction of $\mathcal{C}_{\text{b}}$ under u_{b}.

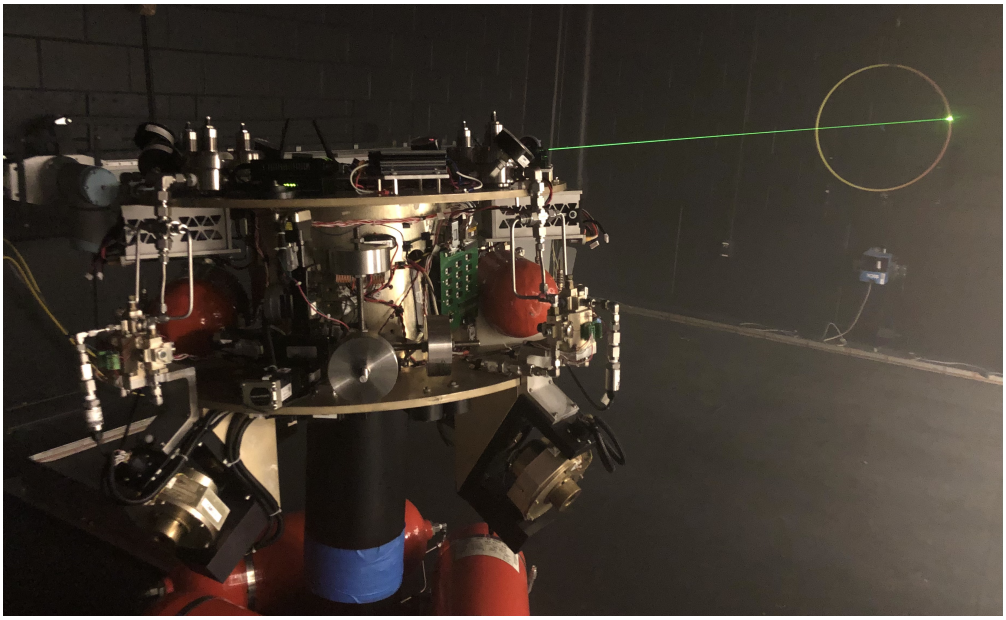


Figure 4.4: ASTROS vehicle with laser oriented along the boresight direction $\widehat{b}$ and avoid cone constraint defined such that the laser does not enter the ring.

4.5 Experimental Demonstration on ASTROS Testbed

The RTA is tested in the Autonomous Spacecraft Testing of Robotic Operations in Space (ASTROS) facility at the Georgia Institute of Technology. The spacecraft is controlled with 12 pressurized air thrusters, arranged in a 3-3-3-3 configuration. A linear program is used to allocate the desired control torque to the thrusters. The experiments are conducted with an admissible control set $U = [-0.5,\ 0.5]^3$ Nm and considers disturbance torques in $W = [-0.02,\ 0.02]^3$ Nm.

In the experiment, a 5 degree avoid cone is considered ($\beta_{\max} = 175$ degrees). As shown in Figure 4.4, the avoid cone is constructed in such a way that prevents an onboard laser from entering a ring placed approximately 18 ft away from the vehicle's center of rotation. The primary control input to the vehicle is provided by a human-controlled pilot joystick. The user attempts to orient the vehicle in such a way that the laser enters the ring, however, the RTA mechanism is effective at preventing the safety constraint from being violated. That is, the user has full control of the vehicle when the RTA is inactive, and the RTA activates only as necessary to prevent violations of the LOS constraint.

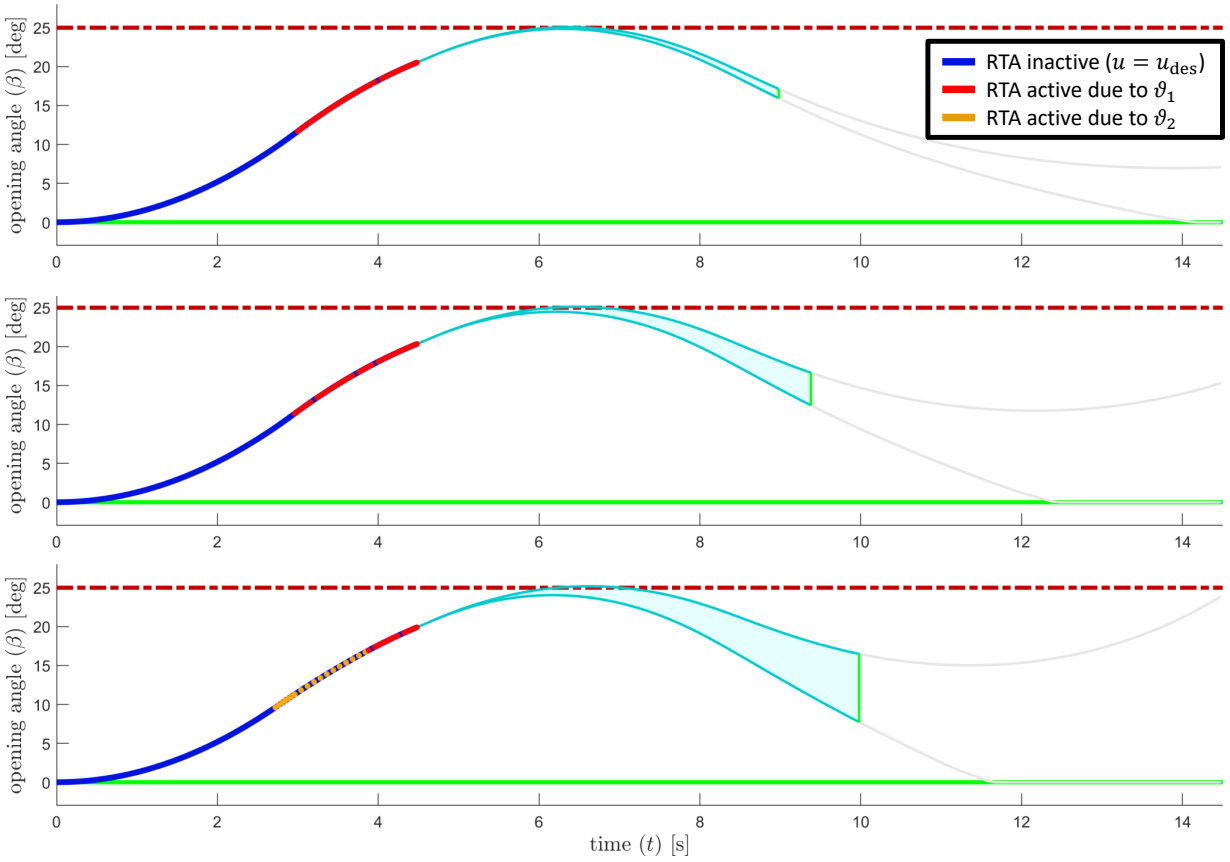


Figure 4.5: Comparison of filtered trajectories and RSO projections under nondeterministic input bounds $w_{\max} = 0.01$ Nm (top), $w_{\max} = 0.03$ Nm (middle), $w_{\max} = 0.05$ Nm (bottom). The vertical green line indicates the time t_{k*} at which the RSO enters $\mathcal{C}_{\text{b}}$.

4.6 Simulated Example

Figure 4.5 shows a simulation of the ASTROS system under a step input $u_{\text{des}} = [0.5, 0.5, 0.5]^T$ Nm, now with $\beta_{\max} = 25$ degrees and perturbation bounds $w_{\max} = 0.01$ Nm (top), $w_{\max} = 0.03$ Nm (middle), $w_{\max} = 0.05$ Nm (bottom). Here it is seen that increasing $w_{\max}$ results in a more conservative filter. Specifically, the RTA activates earlier, and a longer integration time is required before the RSO becomes contained in $\mathcal{C}_{\text{b}}$.

CHAPTER 5

COLLISION-INCLUSIVE PLANNING FOR FREE-FLYING SPACECRAFT

Collisions may be harnessed as a way to improve the overall safety and navigational effectiveness of some spacecraft. However, leveraging this capability in autonomous platforms requires the ability to plan trajectories comprising impulsive contact. This chapter addresses this problem through the development of a collision-inclusive approach to optimal trajectory planning for a three degree-of-freedom free-flying spacecraft. First, experimental data are used to formulate a physically realistic collision model for the spacecraft. It is shown that this model is linear over the expected operational range, enabling a piecewise affine representation of the full hybrid vehicle dynamics. Next, the dynamics model and vehicle constraints are modelled as constraints in a mixed integer program. Experimental comparisons of trajectories with and without collision-avoidance requirements demonstrate the capability of the collision-inclusive strategy to achieve significant performance improvements in realistic scenarios. A simulated case study illustrates the potential for this approach to find damage-mitigating paths in online implementations.

5.1 Motivation

Assistive robotic spacecraft have the potential to enable the automation of many tasks that are not well-suited to be directly performed by astronauts; either because they are too dangerous or overly tedious [138]. Within this context, extravehicular platforms have been proposed for missions such as on-orbit monitoring [139, 140], component assembly [141], and debris removal [142]. Likewise, intravehicular assistive robotics [143] are being developed to fulfill many housekeeping duties inside the International Space Station (ISS). For instance, the Astrobee robot [144], a successor to NASA's highly successful SPHERES [145] testbed, has stated goals of (i) providing a microgravity research platform, (ii) performing mobile camera tasks, and (iii) performing mobile sensor tasks for environment monitoring and inventory management. As with most mobile autonomous platforms,

safe and efficient navigation is key to the successful integration of these vehicles into mission operations.

A popular approach to mobility for microgravity robots is *propulsive free-flying*. This consists of expending propellant to actuate movement between periods of free-flight. One may find key results related to path planning and close proximity operations for this case in [146, 147, 148, 101]. However, a common issue facing this approach is that propellant is often expensive to acquire or in limited supply. As a consequence of this, fuel-efficiency has become one of the primary performance characteristics for spacecraft. The desire to reduce costs has motivated the development of two alternate navigation modalities. The first is zero-g climbing [149], where the vehicle uses grasping contact in the surrounding environment to traverse between locations. A proposed faster and simpler alternative is the hopping modality [150, 151]. In this case, the vehicle uses a robotic arm to propel itself between some fixed handrails. While this strategy is attractive in the sense that it is completely propellantless, it is also more restrictive than the propulsive free-flying strategy in the sense that it requires the precise coordination of a robotic arm and requires handrails to be present over the operational region.

This chapter presents a new approach to mobility for assistive spacecraft: supplementation of propulsive free-flying with planned collisional contact (bouncing). It is shown that this offers a strategy that is both less restrictive than the propellant-free approaches, and often more efficient than its collision-free counterpart. In contrast to hopping, where a robotic arm interacts with the environment to provide the energy needed to change the momentum of the spacecraft, bouncing achieves similar maneuvers passively though impulsive contact. For example, a spacecraft needing to redirect itself inside a corridor may do so swiftly with a single well planned collision, rather than executing the series of maneuvers needed for coordinated hopping. Since the interaction is passive, bouncing poses very little requirements on the vehicle or the surrounding environment itself. Hence the main challenge stems from the task of developing an effective motion planning strategy to leverage this capability. Focusing specifically on the case of small, assistive intravehicular spacecraft, it is assumed that the vehicle operates in the proximity of fixed surfaces with which it

may collide, and that both the vehicle and the surface are able to withstand low-speed impact.

There is a rich body of work related to impulsive contact in robotics, spanning applications such as: running [152]; jumping [153]; batting [154]; air hockey [155]; and car following [156], to name a few. In addition, the problem appears in the aerospace context, within landing [157], docking [158], grasping [159], and bouncing on planetary bodies [160]. Looking specifically at the case of vehicle collisions, there has been foundational work in analyzing the stability and robustness of a colliding vehicle [161], designing vehicles that are tolerant to collisions [162], and even extracting localization information from instances of impact [163]. Collisions can further be harnessed as a practical means of improving the effectiveness of trajectories. Through dissipation of energy or redirection of momentum, colliding agents are endowed with greater maneuverability. One can observe many examples of this phenomenon in competitive situations *e.g.* swimming, parkour, or in nature *e.g.* animals pushing off of [164] or jumping between objects. However, the use of planned impulsive contact explicitly for performance gains has only recently been considered in the context of robot trajectory planning. In [165], the authors use a mixed integer linear programming (MILP) formulation to derive a time-optimal trajectory incorporating planned collisions for a point mass. In this chapter, these initial results are utilized to develop a collision-inclusive, optimal trajectory-planning formulation for in-plane motion of a free-flying spacecraft. Note that since the overall set of trajectories allowing collisions encompasses all collision-free trajectories as well, the optimal performance with respect to any objective function must either remain the same or improve when compared to the case where collisions are always avoided.

In addition to performance benefits, collisions may be utilized to improve the safety of a vehicle in the presence of observed changes in the surrounding environment. Intuitively: in situations where collisions cannot be avoided, a safest plan of action incorporating the collision may be found. Looking specifically at the case of online model predictive control (MPC), hard collision-avoidance constraints may render the problem infeasible when collisions are unavoidable. This problem can be addressed by either resorting to a backup controller when the MPC is not feasible [166] or softening the constraints (*i.e.* replacing constraints with penalties in the objective function)

such that feasibility is preserved [167]. This prototypical constraint-softening approach is extended with the addition of an explicit model of the collision dynamics formulated in the constraints. In addition to remaining feasible in the presence of an inevitable collision, this allows the vehicle to plan around the collision, all while minimizing a penalty function that captures the estimated damage cost. This additional safety measure may offer a particularly useful tool for platforms proposing autonomous operation in the presence of humans.

The remainder of the chapter is outlined as follows: In Section 5.2, the mathematical preliminaries required to develop the main results are reviewed. In Section 5.3.1 the assumptions on the spacecraft are introduced and basic dynamical constraints are developed. Experimental collision data is obtained and used to derive a realistic collision model for the spacecraft in Section 5.3.2. Section 5.4 uses the motion model and vehicle constraints to specify an optimal strategy for moving between states. In Section 5.5.1 an experimental case study is described, and the performance of the collision-inclusive algorithm is compared to that of the collision-free case. It is shown that the proposed method is capable of significantly reducing a chosen objective function. Finally, Section 5.5.2 explores potential safety applications with a simulated scenario.

5.2 Representation of Hybrid Systems with Mixed Integer Constraints

MIP allows for the representation of hybrid systems by associating integer variables with the current mode of the system. Specifically, integer variables (also known as event variables) allow for the direct expression of first order logic over the constraints. These variables may be assigned a unique value based on the location of the state vector, and in turn be used to relax a different set of constraints over the continuous variables. To demonstrate this, let us consider the following case where an inequality condition $c^T z < d$ is used to *activate* distinct equality constraints,

$$\begin{cases} {a_0}^T z = b_0 & \text{if } c^T z < d \\ {a_1}^T z = b_1 & \text{if } c^T z \geq d, \end{cases} \tag{5.1}$$

with $z,\ a_i,\ c \in \mathbb{R}^n,\ b_i,\ d \in \mathbb{R},\ i = 0, 1$. The main tools at our disposal for representing hybrid systems as programs in the form of Eq. (5.1) come from the lemmas below, which define relationships between implications and inequalities of real and binary decision variables. Let $\zeta \in \{0, 1\}$, $z \in Z \subset \mathbb{R}^n$, and parameters $a, c \in \mathbb{R}^n$, $b, d \in \mathbb{R}$. The following results are noted [118, 25, 168, 8]

Lemma 5.2.1. (see [3]) *Given* $M \in \mathbb{R}$ *such that* $\max_{z \in Z}(d - c^T z) < M$*, the following are equivalent:*

$$(i)\ [c^T z < d] \implies [\zeta = 1] \qquad (ii)\ c^T z + M\zeta \geq d$$

$\Diamond$

Proof. If $\max_{z \in Z}(d - c^T z) < M$ holds, then the statement (ii) is true for all $z \in Z$ when $\zeta = 1$. Given $\zeta = 0$, (ii) is true when $c^T z < d$ holds, and is false otherwise. Thus, the truth values for (ii) are identical to the implication (i) for all assignments. $\square$

Lemma 5.2.2. (see [3]) *Given* $M \in \mathbb{R}$ *such that* $\max_{z \in Z}(c^T z - d) < M$*, the following are equivalent:*

$$(i)\ [\zeta = 1] \implies [c^T z < d] \qquad (ii)\ c^T z - M(1 - \zeta) < d$$

$\Diamond$

Proof. If $\max_{z \in Z}(c^T z - d) < M$ holds, then (ii) is true for all $\zeta = 0$. Given $\zeta = 1$ (ii) is equivalent to $c^T z < d$. $\square$

The proofs for the above Lemmas may also be made apparent via the form of a truth table. Note that these can be applied together to form an equivalency. Likewise, application to inequalities of opposing sense (in conjunction) extends the result to the case of equality constraints.

Lemma 5.2.3. (see [4], Sec. 16.4) *Given* $M \in \mathbb{R}$ *such that* $\max_{z \in Z}(a^T z - b) < M$*, the following are equivalent:*

$$(i)\ [\zeta = 1] \implies [a^T z = b] \qquad (ii)\ [a^T z - M(1 - \zeta) \leq b]\ \wedge\ [a^T z + M(1 - \zeta) \geq b]$$

$\Diamond$

Proof. If $\max_{z \in Z}(a^T z - b) < M$ holds, then (ii) is trivially satisfied for $\zeta = 0$. Given $\zeta = 1$, (ii) is equivalent to $a^T z = b$. Equivalence then follows from the truth table. $\square$

From here these results may be combined to yield a conjunction of mixed integer inequalities that is equivalent to Eq. (5.1) over some specified range on z,

Theorem 5.2.1. (see [4], Sec. 16.4) *Given $M \in \mathbb{R}$ sufficiently large such that $\max(\max_{z \in Z}(|c^T z - d|), \max_{z \in Z}(a^T z - b)) < M$ holds, then the system* (5.1) *is equivalent to:*

$$[{a_0}^T z - M\zeta \leq b_0] \quad \wedge \quad [{a_1}^T z - M(1-\zeta) \leq b_1] \quad \wedge \quad [c^T z - M\zeta < d]$$
$$[{a_0}^T z + M\zeta \geq b_0] \quad \wedge \quad [{a_1}^T z + M(1-\zeta) \geq b_1] \quad \wedge \quad [c^T z + M(1-\zeta) \geq d]$$

$\Diamond$

In practice, the parameter M should be chosen carefully. While values that are too low may not satisfy the above conditions, excessively large values will decrease computational efficiency, and may introduce numerical error. For notational simplicity, the sequel uses the same parameter M in all instances of this method. Note that from a computational viewpoint it is often better to avoid strict inequalities in implementation. This may be accomplished by using a non-strict inequality and adding a small number ε to the side with lesser value.

5.3 Vehicle Description and Constraints

The system of interest consists of a single free-flying spacecraft in the presence of one or more flat surfaces. The spacecraft is subject to input constraints and is restricted to moving within a plane. The colliding surface is assumed to be fixed and orthogonal to this plane. The collision model is developed empirically for the specific use case of operation on the robot and testbed at Stanford's Space Robotics Facility (see Figure 5.1). A detailed description of this environment and the experimental setup is provided in Section 5.5.1.

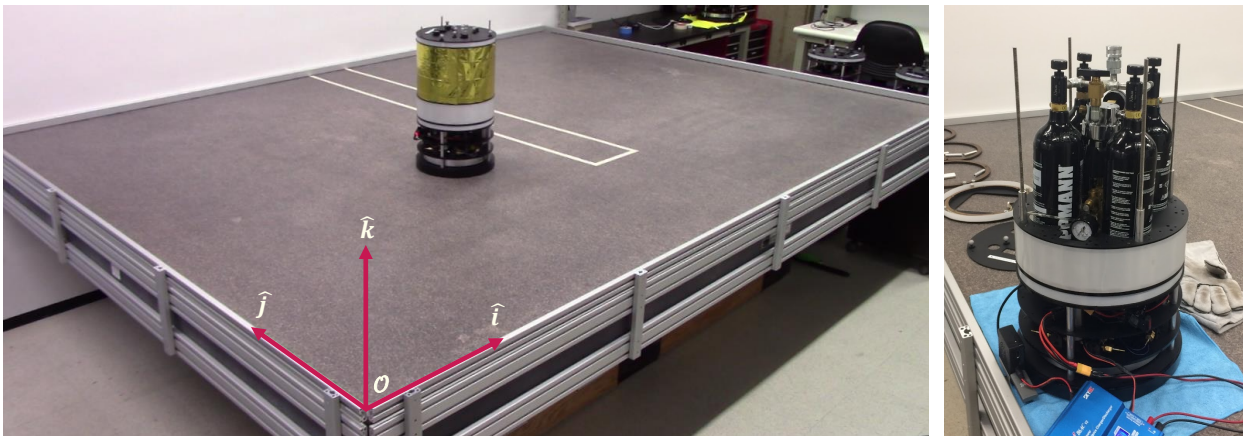


Figure 5.1: Free-Flyer spacecraft and testbed.

5.3.1 Free Flight Dynamics

The motion of the spacecraft is expressed in an inertial frame $\mathcal{F}_{\mathrm{I}} = (\mathcal{O}, \hat{i}, \hat{j}, \hat{k})$ with right-handed orthogonal basis vectors $\hat{i}$, $\hat{j}$ lying in the plane of motion and $\hat{k} = \hat{i} \times \hat{j}$. For the testbed shown in Figure 5.1, the origin $\mathcal{O}$ is taken as the lower left corner, $\hat{i}, \hat{j}$ lie along the testbed boundary, and $\hat{k}$ points upwards. The position of the vehicle's center of mass $\mathcal{O}_{\mathrm{B}}$ with respect to $\mathcal{O}$ is $s = s_{\mathrm{x}}\hat{i} + s_{\mathrm{y}}\hat{j}$ and the translational velocity is $v = v_{\mathrm{x}}\hat{i} + v_{\mathrm{y}}\hat{j}$. A body frame is defined as $\mathcal{F}_{\mathrm{B}} = (\mathcal{O}_{\mathrm{B}}, \hat{i}_{\mathrm{B}}, \hat{j}_{\mathrm{B}}, \hat{k}_{\mathrm{B}})$, with basis vectors $\hat{i}_{\mathrm{B}}$, and $\hat{j}_{\mathrm{B}}$ aligned with the orientation of the thrusters, and $\hat{k}_{\mathrm{B}} = \hat{k}$. The orientation of $\mathcal{F}_{\mathrm{B}}$ with respect to $\mathcal{F}_{\mathrm{I}}$ is θ, with positive increments in the angle θ corresponding to counterclockwise rotations of the spacecraft, as seen from above. This geometry is illustrated in Figure 5.3a. The angular velocity of the vehicle is $\omega = \dot{\theta}\hat{k}$. The *nominal* —*i.e.* collision-free— spacecraft dynamics[1] are then,

$$\ddot{s}_{\mathrm{x}} = u_{\mathrm{x}}, \qquad \ddot{s}_{\mathrm{y}} = u_{\mathrm{y}}, \qquad \ddot{\theta} = u_{\theta} \tag{5.2}$$

where, u_{x}, u_{y} are the translational accelerations due to applied thrust, and u_{θ} is the rotational acceleration from an applied moment, which is generated by changes in the reaction wheel speed from a lower level controller. It is assumed that the thrust inputs are balanced, such that the applied

[1]The modelling and trajectory planning phases will assume a disturbance-free and deterministic spacecraft model. Noise, parametric uncertainty, and errors from approximation are to be addressed through online regulation.

moment comes entirely from the reaction wheel. For the configuration shown in Figure 5.3a, the relationship between translational acceleration and the thrust output from the 8 individual thrusters is,

$$\begin{bmatrix} u_{\mathrm{x}} \\ u_{\mathrm{y}} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \end{bmatrix} u_{\mathrm{T}} \tag{5.3}$$

where m is the mass of the spacecraft and $u_{\mathrm{T}} \in [0,\, u_{\mathrm{T,max}}]^8$ is the vector of individual thruster output forces. Here $u_{\mathrm{T,max}}$ represents the maximum force output of a single thruster.

Note that the thruster arrangements on the spacecraft are such that the maximum accelerations achievable in the $\widehat{i}$, $\widehat{j}$ directions are functions of the body orientation θ. This can be simplified with the use of a conservative inner approximation on the maximum acceleration from thrust $u_{\max}(\theta)$, which generates a condition that is uniform (not dependent on orientation) in the inertial frame.

$$u_x^2 + u_{\mathrm{y}}^2 \leq u_{\max}^2, \qquad u_{\max} = \min_{\theta}(u_{\max}(\theta))) \tag{5.4}$$

With the present geometry, $u_{\max} = \frac{2}{m} u_{\mathrm{T,max}}$.

5.3.2 Collision Model

In order to develop a framework for optimizing trajectories that allow collisions, it is necessary to first develop a model for the collision effects on the spacecraft. Collisions are generally difficult to understand and model conceptually, as a first principles analysis requires the consideration of many interacting physical phenomena relating to the geometric, material, and inertial properties of each body involved; many of which are in themselves difficult to model accurately. Many approaches that have been proposed to model *general* collision behavior over a wide range of scenarios [169, 170]. However, since the present case involves a specific pair of objects over a relatively limited range of conditions, it is desirable to develop an algebraic collision model empirically, by directly considering the relationship between the velocities immediately before and after the instant of contact with no thrust commanded. Figure 5.2 shows the effects of 82 individual collisions for the free-flyer spacecraft and testbed shown in Figure 5.1 . Within the tested range, the data suggest

that the changes in rotational velocity ($\Delta\omega$), translational velocity normal to the wall (Δv_{N}) and tangent to the wall (Δv_{T}), all follow a linear relationship with the pre-collision normal velocity (v_{N}^-) and relative velocity of the point of contact (v_{rel}^-). Furthermore, it is observed that for this set of parameters, effects in the normal direction are uncoupled from the tangential and rotational effects, leading to the following model,

$$\begin{bmatrix} \Delta v_{\mathrm{T}} \\ \Delta v_{\mathrm{N}} \\ \Delta\omega \end{bmatrix} = \begin{bmatrix} 0 & \kappa_{\mathrm{T}} \\ \kappa_{\mathrm{N}} & 0 \\ 0 & \kappa_{\omega} \end{bmatrix} \begin{bmatrix} v_{\mathrm{N}}^- \\ v_{\mathrm{rel}}^- \end{bmatrix} \tag{5.5}$$

where,

$$v_{\mathrm{rel}}^- := v_{\mathrm{T}}^- + R\,\omega^- \tag{5.6}$$

$(\kappa_{\mathrm{T}},\, \kappa_{\mathrm{N}},\, \kappa_{\omega}) = (-0.29, -1.43, -5.0)$, and R is the radius of the spacecraft, measured from the outer rim to the rotation center. The coefficients are obtained via a least squares regression on the model error.

If it is assumed that the collision occurs instantaneously, the positions after the collision can be obtained by integrating the equations of motion with pre-collision velocities until the point of contact, and post-impact velocities afterward. Let $\Delta t := \Delta t^- + \Delta t^+$ be the period between the state measurements, and δ be the effective location of the wall along the orthogonal axis (inflated by R, as seen in Figure 5.3b). Then the experimental model of Eq. (5.5) yields the following position update equations,

$$\begin{aligned} \Delta s_{\mathrm{T}} &= (1+\kappa_{\mathrm{T}})\Delta t v_{\mathrm{T}}^- + \kappa_{\mathrm{T}} R \Delta t \omega^- - \kappa_{\mathrm{T}}(v_{\mathrm{T}}^- + R\omega^-)\Delta t^- \\ \Delta s_{\mathrm{N}} &= (1+\kappa_{\mathrm{N}})\Delta t v_{\mathrm{N}}^- + \kappa_{\mathrm{N}}(s_{\mathrm{N}}^- - \delta) \\ \Delta\theta &= (1+\kappa_{\omega} R)\Delta t \omega^- + \kappa_{\omega}\Delta t v_{\mathrm{T}}^- - \kappa_{\omega}(v_{\mathrm{T}}^- + R\omega^-)\Delta t^- \end{aligned} \tag{5.7}$$

where the time until collision is,

$$\Delta t^- = \frac{(\delta - s_{\mathrm{N}}^-)}{v_{\mathrm{N}}^-} \tag{5.8}$$

Note that the term Δt^- introduces a nonlinearity in the tangential and rotational update laws. Making the approximation that collision occurs midway through the interval $\Delta t^- = 0.5\Delta t$ allows us to obtain a linear form of these equations. The bounds on error from this assumption can be calculated from the maximum difference between the exact and approximated equations, which yields,

$$e_{\mathrm{T}} \leq \frac{\kappa_{\mathrm{T}}}{2}|v_{\mathrm{rel}}^-|\Delta t\,, \qquad e_{\mathrm{N}} = 0\,, \qquad e_\theta \leq \frac{\kappa_\omega}{2}|v_{\mathrm{rel}}^-|\Delta t\,, \tag{5.9}$$

where e_{T}, e_{N}, e_θ are the errors in the tangential, normal, and angular directions respectively. Note that errors vanish both as $|v_{\mathrm{rel}}^-|$ decreases, and for finer resolutions Δt. The collision geometry of the free-flyer system is illustrated in Figure 5.3b.

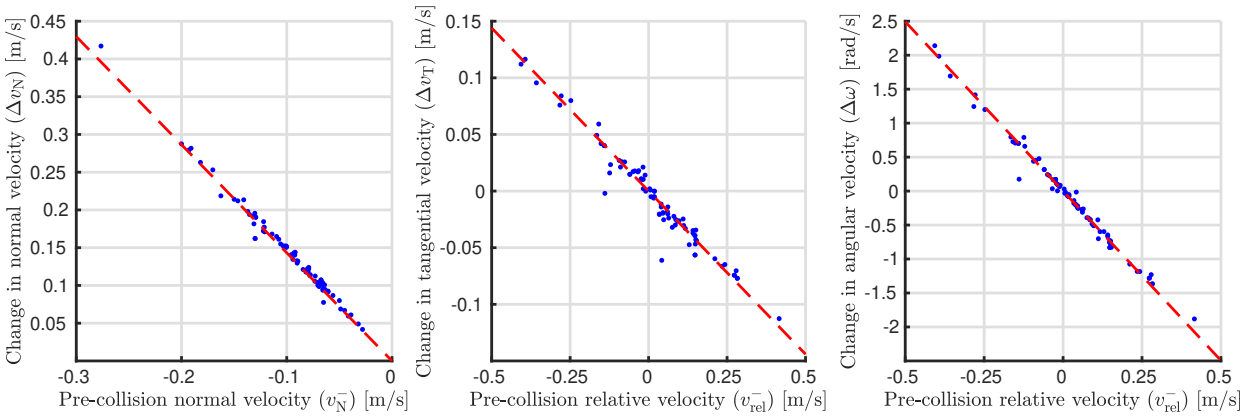


Figure 5.2: Observed data from 82 collisions, with linear interpolations taken with respect to the least squares error.

5.4 Problem Formulation

This section formulates the problem of generating optimal trajectories for a spacecraft in the presence of (i) an obstacle avoidance region $\mathcal{A}$, composed of N_{P} convex polygons $\mathcal{P}_k$, with $k \in \{1, \ldots, N_{\mathrm{P}}\}$, and (ii) surfaces $\mathcal{S}$, $\bar{\mathcal{S}}$ (representing half-planes and convex polygons respectively) with which collisions are permissible. For both $\mathcal{P}$ and $\mathcal{S}, \bar{\mathcal{S}}$, the conventions used are such that the *interior* of the walls is denoted by the union of *sub*-level surfaces of some defined planes

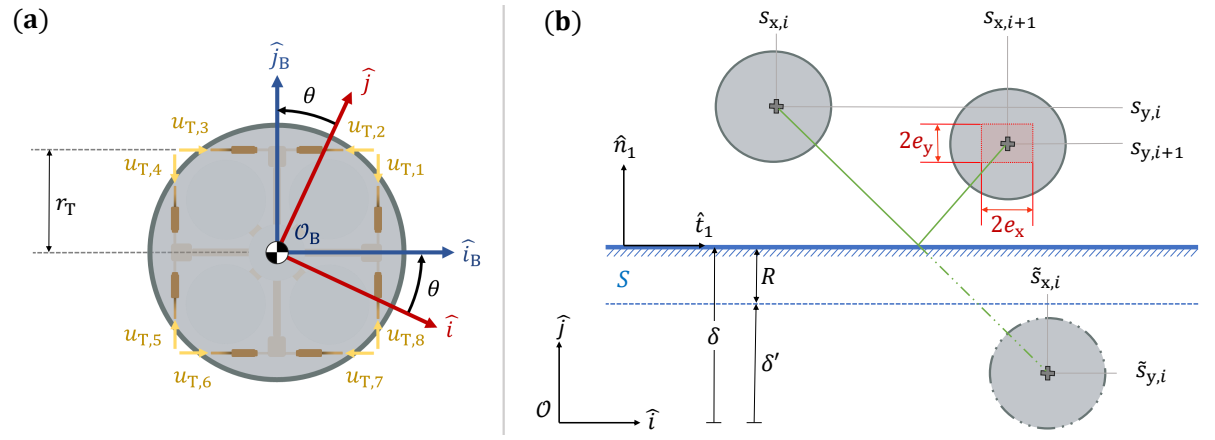


Figure 5.3: Geometry and conventions for (a) freeflyer spacecraft, (b) collision with a flat wall $\mathcal{S}$.

in $\mathbb{R}^3$, and the exterior is the complement of the interior. It is shown that the combined dynamics of the spacecraft, saturation constraints, and obstacle avoidance conditions are all amenable to approximation with piecewise affine constraints. In light of this, the trajectory optimization problem is posed as a MIP. The discrete time approximations of the models developed in previous sections are considered over a horizon of $i = 1, \ldots, \tau$. The state of the vehicle at the i^{th} time-step is defined as $x_i^T = [s_{x,i},\, s_{y,i},\, \theta_i,\, v_{x,i},\, v_{y,i},\, \omega_i]$, and control vector as $u_i^T = [u_{x,i},\, u_{y,i},\, u_{\theta,i}]$. For completeness, some basic control and obstacle avoidance constraint formulations found in [25, 8, 100] are expounded upon.

5.4.1 Obstacle Avoidance and Saturation Constraints

The saturation constraint Eq. (5.4) can be represented by approximating the Euclidean norm with an N_U sided polygon,

$$u_{x,i} \sin\left(\frac{2\pi n}{N_U}\right) + u_{y,i} \cos\left(\frac{2\pi n}{N_U}\right) \leq u_{\max}, \quad n = 1, \ldots, N_U, \quad i = 1, \ldots, \tau. \tag{5.10}$$

While the approximation improves with the number of sides N_U, the added constraints may increase the amount of time required to calculate the solution. The aggregate obstacle avoidance

region $\mathcal{A}$ can be constructed from a set of N_{P} convex polygons $\mathcal{P}_k$,

$$\mathcal{A} := \{z \in \mathbb{R}^2 \mid \bigvee_{k=1}^{N_{\mathrm{P}}} z \in \mathcal{P}_k\} \tag{5.11}$$

where

$$\mathcal{P}_k := \{z \in \mathbb{R}^2 \mid c_{k,q}^T z < d_{k,q},\ q = 1, \dots, N_{\mathrm{Q},k}\} \tag{5.12}$$

where $c_{k,q} \in \mathbb{R}^2$, $d_{k,q} \in \mathbb{R}$ specify the q^{th} side of the k^{th} polygon, which has $N_{\mathrm{Q},k}$ sides. The avoidance constraint $s \notin \mathcal{A}$ can be constructed by defining event variables $\psi_{k,q,i} \in \{0, 1\}$ such that $c_{k,q}^T s_i < d_{k,q} \implies \psi_{k,q,i} = 1$, and ensuring that the position of the vehicle lies in the positive end (exterior) of at least one of half-spaces defining the walls of each polygon $\mathcal{P}_k$. This is accomplished with the following constraints,

$$\begin{aligned} &\bigwedge_{q=1}^{N_{\mathrm{Q},k}} c_{k,q}^T s_i \geq d_{k,q} - M\psi_{k,q,i} \\ &\sum_{q=1}^{N_{\mathrm{Q},k}} \psi_{k,q,i} \leq N_{\mathrm{Q},k} - 1, \end{aligned} \qquad k = 1, \dots, N_{\mathrm{P}}, \quad i = 1, \dots, \tau. \tag{5.13}$$

Note that each conjunct is an application of Lemma 5.2.1, and the summations enforce the condition that there is at least one side q in each polygon such that $c_{k,q}^T s_i \geq d_{k,q}$.

Example 5.4.1. Rectangular Boundary Consider the simplified case of a rectangle $\mathcal{A} = \mathcal{P}_1 = \{z \in \mathbb{R}^2 \mid z_1 \in (z_1^{\min}, z_1^{\max}),\ z_2 \in (z_2^{\min}, z_2^{\max})\}$. The equivalent MIP constraints for the condition

$s \notin \mathcal{A}$ are,

$$
\begin{aligned}
-s_{\mathrm{x}} + M\psi_1 &\geq -z_1^{min} \\
s_{\mathrm{x}} + M\psi_2 &\geq z_1^{max} \\
-s_{\mathrm{y}} + M\psi_1 &\geq -z_2^{min} \\
s_{\mathrm{y}} + M\psi_2 &\geq z_2^{max} \\
\sum_{q=1}^{4} \psi_q &\leq 3.
\end{aligned}
$$

■

5.4.2 Representing Dynamics in the Presence of a Single Collision Surface

For notational simplicity, it is assumed for this case that the basis vectors of $\mathcal{F}_I$ are oriented with the wall $\mathcal{S}$ so that $\widehat{j}$ points away from the wall, $\widehat{i}$ is tangent, and $\widehat{k} = \widehat{i} \times \widehat{j}$ remains pointed upwards (see Figure 5.3b). The discrete time equations of motion are given by,

$$
x_{i+1} - x_i = \begin{cases} Ax_i + Bu_i & \text{if} \quad \zeta_{i+1} = 0 \\ A_{\mathrm{c}} x_i + b_{\mathrm{c}} & \text{if} \quad \zeta_{i+1} = 1 \,, \end{cases} \qquad i = 1, \ldots, \tau - 1 \tag{5.14}
$$

where A, B represent the nominal dynamics,

$$
A = \begin{bmatrix} 0_3 & I_3 \Delta t \\ 0_3 & 0_3 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.5 I_3 \Delta t^2 \\ I_3 \Delta t \end{bmatrix} \tag{5.15}
$$

and A_{c}, b_{c} represent the collision dynamics,

$$A_{\mathrm{c}} = \begin{bmatrix} 0 & 0 & 0 & (1+0.5\kappa_{\mathrm{T}})\Delta t & 0 & 0.5\kappa_{\mathrm{T}} R\Delta t \\ 0 & \kappa_{\mathrm{N}} & 0 & 0 & (1+\kappa_{\mathrm{N}})\Delta t & 0 \\ 0 & 0 & 0 & 0.5\kappa_{\omega}\Delta t & 0 & (1+0.5\kappa_{\omega}R)\Delta t \\ 0 & 0 & 0 & \kappa_{\mathrm{T}} & 0 & \kappa_{\mathrm{T}} R \\ 0 & 0 & 0 & 0 & \kappa_{\mathrm{N}} & 0 \\ 0 & 0 & 0 & \kappa_{\omega} & 0 & \kappa_{\omega} R \end{bmatrix}, \quad b_{\mathrm{c}} = \begin{bmatrix} 0 \\ -\kappa_{\mathrm{N}}\delta \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.16}$$

The period between steps i and $i-1$ is Δt, which is assumed to remain constant over all iterations.

Let the wall be located at a distance δ' from the origin of the inertial frame $\mathcal{F}_I$, and define $\delta := \delta' + R$, where R is the radius of the spacecraft, measured from the rotation center to the point of contact. Then the wall can be defined by the set $\mathcal{S} := \{z \in \mathbb{R}^2 \,|\, z_2 < \delta\}$. The occurrence of a collision can be associated with an event variable $\zeta_i \in \{0, 1\}$. This triggers the switch between the nominal and collision dynamics; it is activated (equal to one) on an iteration i if the nominal dynamics predict that the vehicle will enter $\mathcal{S}$ on that iteration, *i.e.*

$$s_{\mathrm{y},i} + v_{\mathrm{y},i}\Delta t + u_{\mathrm{y},i} 0.5\Delta t^2 < \delta \iff \zeta_{i+1} = 1, \qquad i = 1, \dots, \tau - 1 \tag{5.17}$$

Using Lemmas 5.2.1, 5.2.2, Eq. (5.17) can be expressed with the equivalent set of constraints,

$$\begin{aligned} s_{\mathrm{y},i} + v_{\mathrm{y},i}\Delta t + u_{\mathrm{y},i} 0.5\Delta t^2 + M\zeta_{i+1} &\geq \delta \\ s_{\mathrm{y},i} + v_{\mathrm{y},i}\Delta t + u_{\mathrm{y},i} 0.5\Delta t^2 - M(1-\zeta_{i+1}) &< \delta \end{aligned} \qquad i = 1, \dots, \tau - 1 \tag{5.18}$$

and from Theorem 5.2.1, it is clear that Eq. (5.14) is equivalent to,

$$\begin{aligned} x_{i+1} - x_i - Ax_i - Bu_i + MI_6\zeta_{i+1} &\geq 0 \\ x_{i+1} - x_i - Ax_i - Bu_i - MI_6\zeta_{i+1} &\leq 0 \\ x_{i+1} - x_i - A_{\mathrm{c}}x_i - b_{\mathrm{c}} + MI_6(1-\zeta_{i+1}) &\geq 0 \\ x_{i+1} - x_i - A_{\mathrm{c}}x_i - b_{\mathrm{c}} - MI_6(1-\zeta_{i+1}) &\leq 0, \end{aligned} \qquad i = 1, \dots, \tau - 1. \tag{5.19}$$

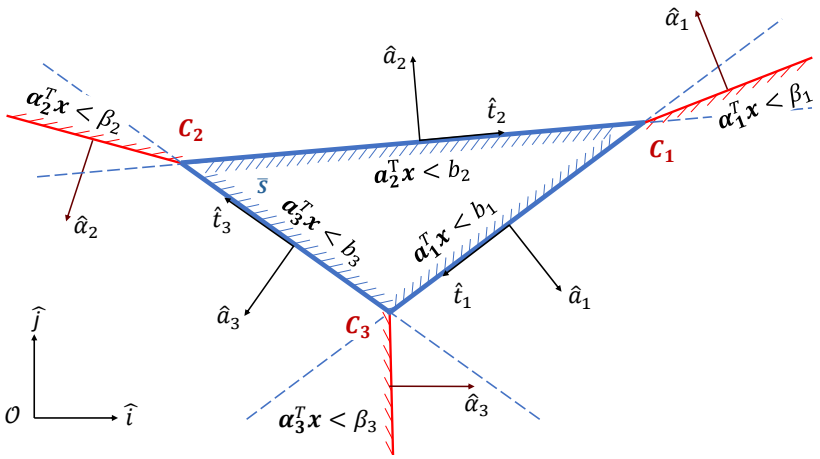


Figure 5.4: Example geometry for triangular collision polygon $\bar{\mathcal{S}}$.

Note that collisions may have the undesirable effect of imparting an external moment onto the spacecraft. While this may be useful for translating stored angular momentum into lateral momentum in safety critical scenarios, it could also lead to saturation of the reaction wheels over time. As such, it may be desirable to minimize momentum transfer by constraining the relative velocity of the contact point to zero at the time of collision: $\zeta_i = 1 \implies v_{\text{rel},_i} = 0$. Equivalently, from Lemma 5.2.3,

$$\begin{aligned} v_{\text{x},i} + R\omega_i - M(1-\zeta_i) \leq 0 \\ v_{\text{x},i} + R\omega_i + M(1-\zeta_i) \geq 0 \end{aligned} \qquad i = 2, \dots, \tau. \tag{5.20}$$

Note that meeting this condition preserves the initial tangential and angular velocity over the collision.

5.4.3 Representing Dynamics in the Presence of Polygonal Collision Surfaces

At the expense of introducing some complexity, collision surfaces $\mathcal{S}$ can be generalized from half-planes to convex polygons,

$$\bar{\mathcal{S}} := \{z \in \mathbb{R}^2 \,|\, a_j^T z < b_j \,,\; j = 1, \dots, N_{\bar{\mathcal{S}}}\} \tag{5.21}$$

where $N_{\bar{\mathcal{S}}}$ is the number of sides in the polygon and the indices j label the walls in a counterclockwise order. In contrast to the previous case, the basis vectors $\widehat{i}$, $\widehat{j}$ are not restricted to a particular orientation. It is assumed that the vehicle collides into the j^{th} wall of $\bar{\mathcal{S}}$ on iteration i if: (i) the position at timestep $i-1$ is closest to the boundary of the j^{th} wall, and (ii) the nominal update equation predicts that the vehicle will enter the interior of $\bar{\mathcal{S}}$ on iteration i. It is convenient to represent the second condition as $\tilde{s}_i := A_{\mathrm{s}} x_{i-1} + B_{\mathrm{s}} u_{i-1} \in \bar{\mathcal{S}}$, where $A_{\mathrm{s}},\ B_{\mathrm{s}}$ are the first two rows of $A,\ B$, corresponding to the position update under the nominal dynamics. Likewise the first condition can be represented as $s_{i-1} \in \mathcal{C}_j$ where $\mathcal{C}_j$ is the region exterior to the polygon, closest to wall j. This can be defined as,

$$\mathcal{C}_j := \{ z \in \mathbb{R}^2 \,|\, \alpha_j^T z < \beta_j \,,\ \alpha_{\sigma(j)}^T z \geq \beta_{\sigma(j)} \,,\ a_j^T z \geq b_j \,\} \tag{5.22}$$

where $\sigma(j)$ is the j^{th} element of $\sigma := (N_{\bar{\mathcal{S}}}, 1, 2, \ldots, N_{\bar{\mathcal{S}}} - 1)$, and $\alpha_j,\ \beta_j$ define the half-space bisecting wall j and the next wall in the counterclockwise rotation, such that $\alpha_j^T z < \beta_j$ is satisfied for points closer to the j^{th} edge. An example configuration is shown in Figure 5.4.

The goal is to define event variables $\Xi_{i,j} \in \{0, 1\}$ to indicate the occurrence of collisions. Specifically, it is desired that $\Xi_{i,j} = 1$ when a collision occurs with the j^{th} wall on iteration i,

$$\Xi_{i,j} = 1 \iff \tilde{s}_i \in \bar{\mathcal{S}} \,\wedge\, s_{i-1} \in \mathcal{C}_j, \qquad i = 2, \ldots, \tau, \quad j = 1, \ldots, N_{\bar{\mathcal{S}}}. \tag{5.23}$$

These indicator variables may then be used to activate the collision dynamics for the wall involved in the collision. In order to indicate the position with respect to the walls $(a_j,\ b_j)$ defining the polygon $\bar{\mathcal{S}}$, the following constraints are introduced

$$\begin{aligned} a_j^T \tilde{s}_i + M\gamma_{i,j} &\geq b_j \\ a_j^T \tilde{s}_i - M(1 - \gamma_{i,j}) &< b_j \end{aligned} \qquad i = 2, \ldots, \tau, \quad j = 1, \ldots, N_{\bar{\mathcal{S}}}. \tag{5.24}$$

This fixes $\gamma_{i,j} = 1$ exactly when the constraint $a_j^T \tilde{s}_i < b_j$ is satisfied (*i.e.* $\gamma_{i,j} = 1 \iff a_j^T \tilde{s}_i < b_j$).

Constraints of the same form can be used to indicate the position of the vehicle with respect to the half-spaces defined by $(\alpha_j,\, \beta_j)$,

$$\begin{aligned} \alpha_j^T s_i + M\lambda_{i,j} &\geq \beta_j \\ \alpha_j^T s_i - M(1-\lambda_{i,j}) &< \beta_j \end{aligned} \qquad i = 2, ..., \tau, \quad j = 1, \dots, N_{\bar{\mathcal{S}}} \tag{5.25}$$

which expresses $\lambda_{i,j} = 1 \iff \alpha_j^T \tilde{s}_i < \beta_j$ for the appropriate values of i, j. The equivalencies in Eq. (5.23) can then be enforced by the constraints,

$$\begin{aligned} \sum_{p=1}^{N_{\bar{\mathcal{S}}}} \gamma_{i,p} + \lambda_{i-1,j} - \lambda_{i-1,\sigma(j)} - \gamma_{i-1,j} + M(1-\Xi_{i,j}) &\geq N_{\bar{\mathcal{S}}} + 1 \\ \sum_{p=1}^{N_{\bar{\mathcal{S}}}} \gamma_{i,p} + \lambda_{i-1,j} - \lambda_{i-1,\sigma(j)} - \gamma_{i-1,j} - M\Xi_{i,j} &\leq N_{\bar{\mathcal{S}}} \end{aligned} \tag{5.26}$$

which is applied for $i = 2, \dots, \tau$, $j = 1, \dots, N_{\bar{\mathcal{S}}}$.

The dynamics for this system are then,

$$x_{i+1} - x_i = \begin{cases} Ax_i + Bu_i & \text{if} \quad \sum_{j=1}^{N_{\bar{\mathcal{S}}}} \Xi_{i+1,j} \leq 0 \\ A_{\mathrm{c}}^j x_i + b_{\mathrm{c}}^j & \text{if} \quad \Xi_{i+1,j} = 1, \;\; j = 1, \dots, N_{\bar{\mathcal{S}}}, \end{cases} \qquad i = 1, \dots, \tau - 1 \tag{5.27}$$

where A_{c}^j, b_{c}^j are the collision dynamics for the j^{th} wall. To represent these dynamics, let us first define a local frame for the j^{th} wall $\mathcal{F}_{\mathrm{w}}^j = (\mathcal{O}, \widehat{t}_j, \widehat{a}_j, \widehat{k})$ with $\widehat{t}_j$, $\widehat{a}_j$ pointing tangent and normal to wall j, and $\widehat{k}_j = \widehat{t}_j \times \widehat{a}_j$ upwards. Each local frame is a rotation of $\mathcal{F}_{\mathrm{I}} = (\mathcal{O}, \widehat{i}, \widehat{j}, \widehat{k})$ about $\widehat{k}$ by an angle ϕ_j. Let $L_3(\phi_j) \in \mathbb{R}^{3\times 3}$ be the rotation matrix converting vectors in $\mathcal{F}_{\mathrm{I}}$ to vectors in $\mathcal{F}_{\mathrm{w}}^j$. The collision dynamics for each wall in $\mathcal{F}_{\mathrm{I}}$ can be expressed by rotating the position and velocity vectors to and from this local frame,

$$A_{\mathrm{c}}^j = \Lambda_j A_{\mathrm{c}} \Lambda_j^T \qquad b_{\mathrm{c}}^j = \Lambda_j [0,\; -\kappa_{\mathrm{N}} \frac{b_j}{\|a_j\|_2},\; 0,\; 0,\; 0,\; 0]^T \tag{5.28}$$

where,

$$\Lambda_j := \begin{bmatrix} L_3(\phi_j) & 0_3 \\ 0_3 & L_3(\phi_j) \end{bmatrix} \tag{5.29}$$

The dynamics in Eq. (5.27) are then represented by the following MIP constraints,

$$\begin{aligned} x_{i+1} - x_i - Ax_i - Bu_i - M\Big(\sum_{j=1}^{N_{\bar{\mathcal{S}}}} \Xi_{i+1,j}\Big) \leq 0 \\ x_{i+1} - x_i - Ax_i - Bu_i + M\Big(\sum_{j=1}^{N_{\bar{\mathcal{S}}}} \Xi_{i+1,j}\Big) \geq 0 \qquad j = 1, \ldots, N_{\bar{\mathcal{S}}} \\ x_{i+1} - x_i - A_{\mathrm{c}}^j x_i - b_{\mathrm{c}}^j - M\Xi_{i+1,j} \leq 0 \\ x_{i+1} - x_i - A_{\mathrm{c}}^j x_i - b_{\mathrm{c}}^j + M\Xi_{i+1,j} \geq 0 \end{aligned} \tag{5.30}$$

which are applied at $i = 1, \ldots, \tau - 1$.

5.4.4 Example Objective Functions

In practice, the appropriate choice of an objective function depends on the specific needs of the mission. The proposed methodology does not assume a particular form for the objective. However, since vehicle efficiency is commonly of critical importance to real-world missions, it is useful to review here two common approximations for penalizing actuation using quadratic and linear functions. A simple option is to use the power limiting cost function [119, 120],

$$J_1 = \sum_{i=1}^{\tau} (u_{\mathrm{x},\,i}^2 + u_{\mathrm{y},\,i}^2) \tag{5.31}$$

which forms a Mixed Integer Quadratic Program (MIQP). With the introduction of additional constraints, is also possible to use a PWA approximation of the Euclidean norm of commanded translational acceleration [100]. The resulting cost function is linear,

$$J_2 = \sum_{i=1}^{\tau} G_i\,, \quad s.t. \quad \bigwedge_{n=1}^{N_J} \; G_i \geq u_{\mathrm{x},\,i} \sin\Big(\frac{2\pi n}{N_{\mathrm{J}}}\Big) + u_{\mathrm{y},\,i} \cos\Big(\frac{2\pi n}{N_{\mathrm{J}}}\Big), \quad i = 1, \ldots, \tau. \tag{5.32}$$

The constraints here approximate the second order cone constraints

$$G_i \geq \|[u_{x,i}, u_{y,i}]\|_2, \qquad i = 1, \ldots, \tau \tag{5.33}$$

with an N_{J} sided polygon.

5.5 Applications and Case Studies

The potential benefits of the proposed approach are demonstrated through two case studies. First, the problem formulation developed in Section 5.4 is validated on hardware. The effectiveness of the approach in improving upon a chosen objective function is studied through a comparison between collision-inclusive and collision-free trajectories. Results are tabulated for both ideal and experimental cases. Next, a simulated example provides a qualitative demonstration of how the collision-inclusive planner may be applied in safety-critical applications.

5.5.1 Experimental Performance Comparison

Description of Hardware and Testbed

Experiments were conducted for this work in the Stanford Space Robotics Facility on the free-flyer spacecraft robot testbed. A set of robots is designed to hover frictionlessly on air bearings, thus emulating microgravity dynamics in the plane of a table. Though previous generations of the free-flyer robot used in this experiment operated on compressed air [171], the current iteration of the free-flyer operates on CO_2, owing to CO_2's ability to be stored in liquid form at room temperature at only 1000 psi, resulting in a much higher fuel density than can be achieved at comparable pressures with compressed air. The robots are also equipped with actuators commonly used in spacecraft, namely a reaction wheel for attitude control and 8 cold-gas thrusters primarily for translational control. Due to high capacity of the CO_2 tanks, the robots can perform aggressive thrust maneuvers for over an hour and can hover without thrust for over 10 hours continuously.

The robots use an Odroid XU4 for its primary on-board computation, as well as an mbed

Table 5.1: Free-flyer spacecraft parameters.

Parameter	Value	Unit
average mass (m)	18.08	kg
radius (R)	0.157	m
max individual thruster output ($u_{\mathrm{T,max}}$)	0.20	N
body inertia about spin axis (I_{b})	0.184	kgm^2
reaction wheel inertia (I_{w})	0.029	kgm^2
max acceleration of reaction wheel	0.628	$\mathrm{rad/s}^2$
reaction wheel speed range	[60,340]	RPM

Microcontroller for low-level control of various subsystems. Additionally, the free-flyer software stack is implemented in ROS and is connected to an off-board hub computer, where more heavy computation can be run as needed for planning and control. The ROS stack also gives access to real-time data from a motion-capture system, giving position and velocity information at 120 Hz. The granite table used for experiments is 9' $\times$ 12' —approximately 2.74 m $\times$ 3.66 m— allowing ample room for complex planning scenarios. Further parameters for the free-flyer robot can be found in Table 5.1, where average mass is reported due to variations in the state of the tanks.

Performance Comparison

The spacecraft and testbed described in the above section are considered with $\mathcal{S}$ taken as the lower wall of the testbed and the origin of $\mathcal{F}_{\mathrm{I}}$ at the lower left corner, as shown in Figs. 5.1, 5.5. Performance is compared for vehicles navigating from rest at initial position $s_1 = [0.41\mathrm{m}, 2.29\mathrm{m}]^T$ to rest at final position $s_\tau = [3.15\mathrm{m}, 2.29\mathrm{m}]^T$, while remaining in the boundary of the testbed, and avoiding a central rectangular region $\mathcal{P} = \{z \in \mathbb{R}^2 \,|\, z_1 \in [1.45,\ 2.12],\ z_2 \in [0.57, 2.74]\}$. The cost J_1 introduced in Eq. (5.31) is minimized. The performances of the vehicle are compared both in terms of this approximation, and a fuel cost measured through pulse width modulation (PWM) signals sent to the thruster. Assuming constant mass flow rate through the thrusters, the latter cost is directly proportional to fuel consumption. The relative velocity of the contact point is constrained to zero in order to minimize angular momentum transfer with the wall (Eq. (5.20)). A small penalty on angular velocity is also included to reduce unnecessary spin of the spacecraft, which, due to the

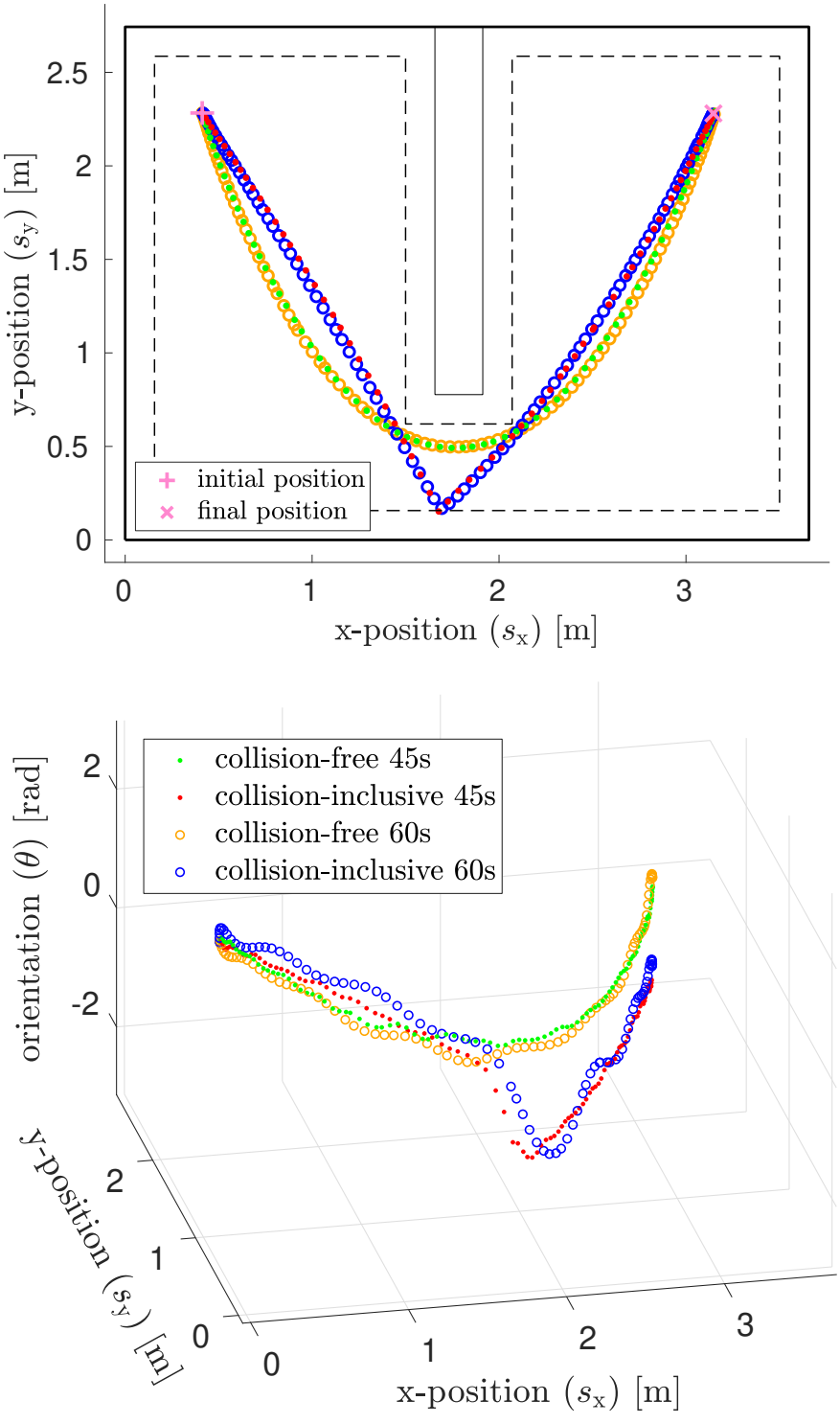


Figure 5.5: Comparison of paths taken in experimental scenarios.

limited update rate of the thrust controller —approximately 2 Hz— may diminish the accuracy of acceleration commands.

The trajectory is generated with Gurobi optimization software using the formulation in Section 5.4 with the parameters listed above. The thrust saturation constraint Eq. (5.4) uses $N_{\mathrm{U}} = 20$ sides in the approximation. To ensure that regulation is possible in the presence of disturbances, the MIP limits the maximum acceleration ($u_{\max}$) to 90% of its theoretical value. The ideal state is tracked using a Linear Quadratic Regulator (LQR) as the ancillary control law. The net control at time $t \in \mathbb{R}$ is,

$$u(t) = u^*(t) + K_{\text{lqr}}(x^*(t) - x(t)) \tag{5.34}$$

where $K_{\text{lqr}} = [2.86\, I_3,\ 14.43\, I_3] \in \mathbb{R}^{3\times 6}$ is the LQR gain matrix, and $u^* \in \mathbb{R}^3$, $x^* \in \mathbb{R}^6$ are the ideal control and state at time t, taken from a polynomial interpolation of the control and state solutions returned from the MIP. The input u is then mapped to PWM signals on the thrusters $u_{\text{pwm}} = u_{\mathrm{T}}/u_{\mathrm{T,max}} \in [0, 1]^8$. This mapping is derived by taking the pseudo-inverse of the mapping from individual thruster forces to forces in the body frame. The resulting mixing equation is balanced in the body frame, ensuring that no moment is produced from the thrusters. An inner PID loop regulates the speed of the reaction wheel, which is used to achieve the desired moment.

Experiments are conducted for this scenario with the time-horizons fixed to 45 and 60 seconds. The experimental update rate of the controller varies slightly from the fixed 0.5s period assumed in the planning phase. As a consequence of this, the 45 s experiments are both completed in 82 steps, and the collision-free and collision-inclusive 60 s experiments are completed in 108 and 109 steps respectively. The trajectories taken are shown in Figure 5.5, and the efficiency measures are plotted against time in Figure 5.6. A video comparison of two experiments may be found in [172]. Note that despite having a significant effect on the total cost, the difference in time allocated to reach the goal has virtually no effect on the shape of the planned path. Table 5.2 shows the total costs for each experiment, along with the corresponding ideal values, and the resulting PWM costs. It is apparent that the collision-inclusive approach is capable of demonstrating significant improvements in overall efficiency for a given time horizon. In particular, reductions in the J_1

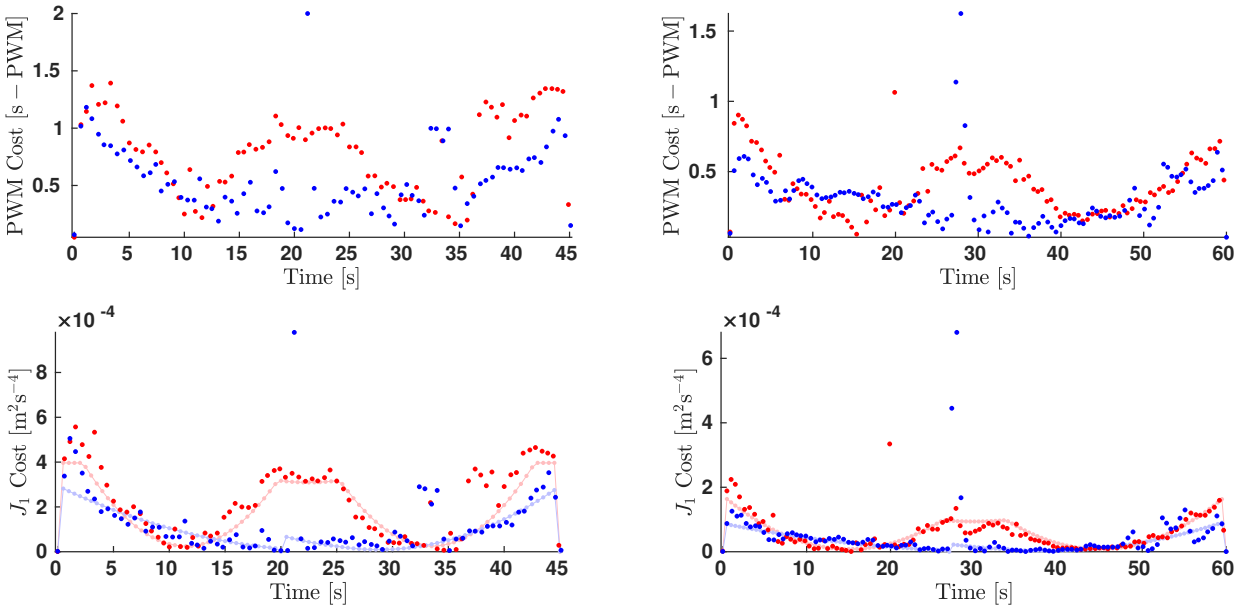


Figure 5.6: Costs vs time for collision-free (red) and collision-inclusive (blue) experiments, and corresponding ideal values (transparent) for the J_1 cost.

cost of 44.3% and 22.9% are seen for the 45 s and 60 s experiments respectively (compared to 47.8%, and 41.2% for the ideal case), and reductions of 31.7% and 23.8% in the PWM cost for the 45 s and 60 s experiments respectively. The main boost in efficiency occurs midway through the trajectory. As the collision-free vehicle requires increased thrust to reduce its velocity and redirect its momentum, the collision-inclusive approach allows the spacecraft to minimize its thrust at this point, gaining the required momentum transfer directly from an impulsive force at the wall. There appears to be some trade-off when using this approach in that a spike in thrust is seen to occur directly after collision. This might be attributed to a number of factors which could potentially lead to increased model error on the collision iteration. For example: sensitivity to modelling the precise location of the wall (δ) or vehicle radius (R); to precisely matching the commanded tangential and angular velocities at the time of collision; or from the zero thrust approximation made in the update equations.

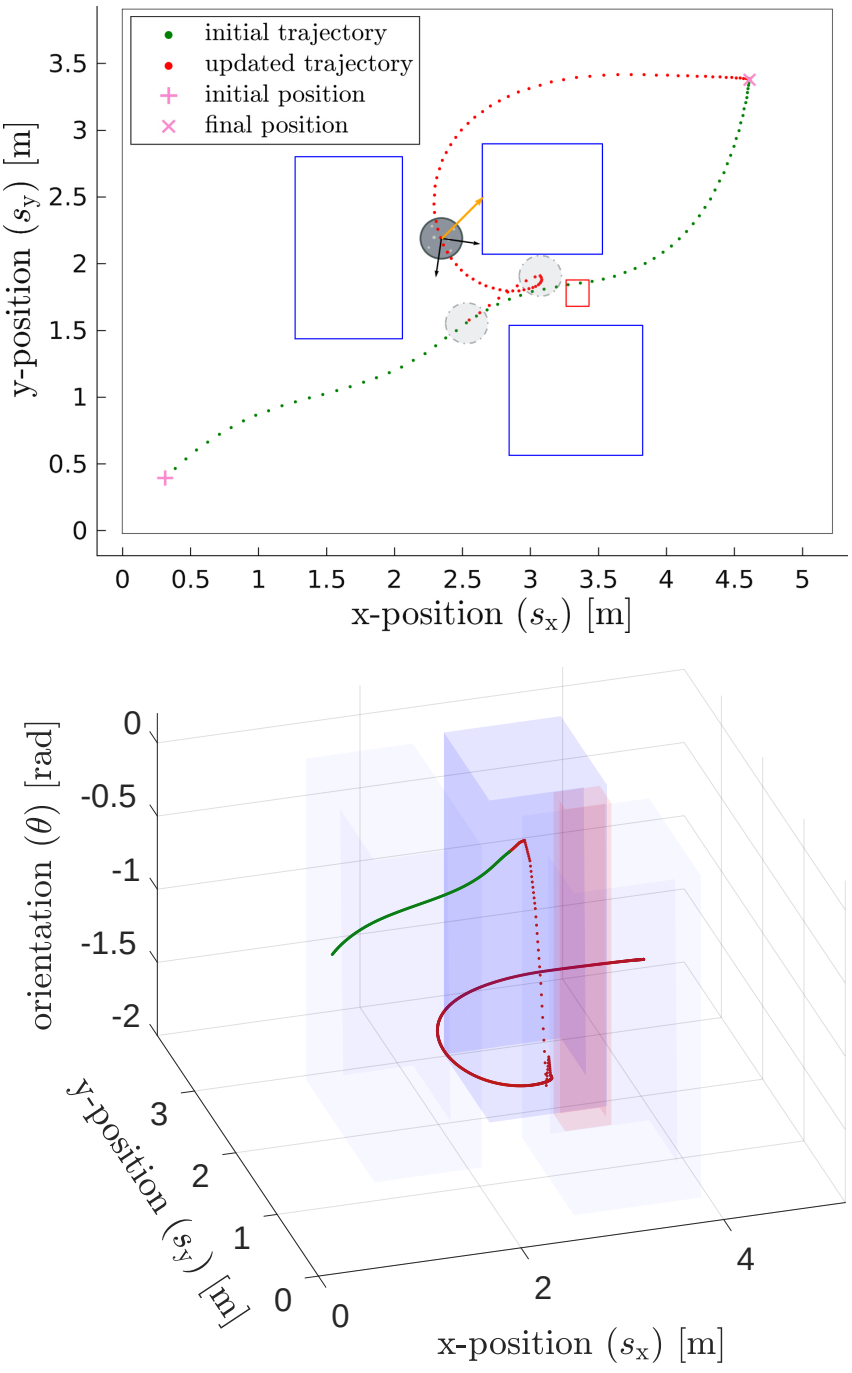


Figure 5.7: Original (green) and updated (red) paths taken to evade observed obstacle (left) [173]; composite trajectory (right).

Table 5.2: Experiment cost values.

Specification	Experimental J_1 Cost $[\mathrm{m}^2/\mathrm{s}^4]$	Ideal J_1 Cost $[\mathrm{m}^2/\mathrm{s}^4]$	Experimental PWM Cost [s]
Collision-Free, 45s	0.01822	0.01619	65.04
Collision-Inclusive, 45s	0.01014	0.00845	44.42
Collision-Free, 60s	0.00633	0.00611	43.76
Collision-Inclusive, 60s	0.00488	0.00359	33.33

5.5.2 Application to Safety-Critical Systems

If collision avoidance is posed as a hard constraint in the problem formulation, then online MPC becomes vulnerable to being rendered infeasible in situations where collisions can no longer be avoided. This situation might arise from a number of factors including model error, external perturbations, or movement of objects in the environment. The problem is exacerbated by the tendency of optimal trajectories to lie near the boundary of the infeasible region (see for example, Figure 5.5). On the other hand, if the constraint to avoid collisions is replaced by a term in the cost function capturing the damage from this event, and the effects of the collision are considered in the constraints, then the planner can not only remain feasible, but direct the vehicle toward an *optimal mitigating action*. A simulated scenario is considered in this section for the purpose of demonstrating the potential of the collision-inclusive planner to bring about enhanced safety in this sense.

Consider the spacecraft and parameters as described in previous sections, now given the task of traversing across a larger, more cluttered environment, consisting of a number of walls whose locations are known to it via an internal map. The vehicle also performs online sensing which it may use to detect unmapped objects in the environment. Collision with the walls is known to cause minor damage to the robot, while collision with a newly detected object is considered more damaging, as neither the type of object nor consequences of hitting it are known in advance. The two strategies for this case are now compared in the environment shown in Figure 5.7. Here the spacecraft (with parameters from previous sections) is given the goal of reaching the point in the top right corner while avoiding obstacles. The vehicle starts on the green trajectory shown in Figure 5.7, however, the vehicle eventually detects the presence of a new obstacle (red box)

obstructing the original path, and is not able to stop in time to prevent collision. If collision avoidance with the obstacles is posed as a hard constraint, then the MPC is rendered infeasible. Without an update, the vehicle may simply continue on its original course and hit the object at high velocity. A more thoughtful implementation might include a backup controller that brings the vehicle to rest as quickly as possible, however even this backup strategy will result in inevitable and uncontrolled collisions with both the wall and obstacle [25].

As an alternate strategy, the presence of the various objects in the program may be incorporated through penalties in a multi-objective cost function. The total cost J will be taken as the sum of some nominal cost J_{nom} (*e.g.* fuel consumption) and a damage cost J_{dam}. Here the damage cost is taken to be a weighted sum of the speeds at the time of impact, defined for the j^{th} wall on the i^{th} iteration as follows,

$$v_{i,j}^{\text{impact}} := -\Xi_{i,j}[v_{\text{x},i},\ v_{\text{y},i}]\frac{a_j}{\|a_j\|_2} \tag{5.35}$$

where $\Xi_{i,j} = 1$ indicates collision with wall defined by a_j on iteration i. Note that impact speed is a quadratic function of the problem variables. Letting σ_1 be the set of indices for the mapped walls (blue) and σ_2 be the indices for the unmapped walls (red). Then the total cost as $J = J_{\text{nom}} + J_{\text{dam}}$ can be expressed with,

$$J_{\text{dam}} := \sum_{i=2}^{\tau}\sum_{j\in\sigma_1} K_1 v_{i,j}^{\text{impact}} + \sum_{i=2}^{\tau}\sum_{j\in\sigma_2} K_2 v_{i,j}^{\text{impact}} \tag{5.36}$$

where, $K_1,\ K_2 \in \mathbb{R}$ weight the collision penalties for each type of object. For this situation, it is specified that K_1 is much less than K_2, directing to vehicle to avoid the unknown object as much as possible. The red path in the Figure 5.7 shows the new trajectory that is calculated using this cost function once the red box is first detected—*i.e.* accounted for in the motion planning. Here the spacecraft is able to leverage the collision dynamics with the blue box to avoid collision with the unknown obstacle altogether. In addition to simply applying thrust to push itself away from the object, the vehicle increases its angular speed before the collision, and utilizes stored angular momentum to push itself away on impact. Additional simulation parameters are listed in [94]. A

video of the simulation may be found at [173].

5.6 Discussion

The formulation may be further developed through consideration of more complex environments or vehicle geometries, either of which may require the construction of a more complex collision model. The results of the trajectory planning may be sensitive to the accuracy of the collision model and other parameters. New regulation strategies may be explored to reduce the loss in efficiency in the time samples surrounding the collision, and to make the strategy more robust to parametric uncertainty. The basic framework for optimizing around collisions may be adapted to applications such as docking or landing, or alternate vehicle platforms such as quad-copters. A key focus of future research will be to expand on the applications toward safety though the creation and implementation of damage minimizing backup controllers. Though the proposed framework is capable of generating the necessary trajectories, the current implementation of the MPC may not be able to generate the aggressive paths quickly enough to be practical for online use. In order to make this more tractable, a learning approach may be utilized to approximate the policy of the collision-inclusive backup controller. Once this has been achieved, the backup strategy may be validated in an experimental scenario. As an alternate safety application, the enhanced maneuverability of the collision-inclusive MPC may be leveraged in a controlled set invariance framework such as the one presented in [174, 23].

If the proposed approach is to be applied in orbit, and over longer timescales, then one may consider replacing the double integrator model with Clohessy-Wiltshire-Hill (CWH) dynamics [101, 175]. Since this model is linear, the nominal dynamics may be incorporated directly into the form presented in Eq. (5.14) and (5.27). In this case, one would either need to modify the collision update equations to account for the new dynamics, or for the errors introduced if the perturbation is neglected between the time steps surrounding the collision. A general extension of the planner to the 6 degree-of-freedom case may potentially be challenging. However, it is important to note that for planar (2-D) and spatial (3-D) operations, the coupling terms between the attitude and

translational dynamics disappear during collision if the relative velocity at the point of contact goes to zero. Hence, once could in principle use this approach to plan trajectories for the translational states, and delegate the task of rotating the spacecraft to match the relative velocity of the wall to a separate, lower-level controller. Similarly, one might simplify the spatial case formulation in a way that allows some of the translation-attitude coupling to be leveraged by constraining the robot to orient its rotation axis to be orthogonal to the direction of relative velocity on collision iterations.

CHAPTER 6
CONCLUSIONS

In order to fully realize the value offered by cyber-physical systems, it is necessary to ensure that these systems avoid taking actions that could be harmful to themselves, their environments, or that would result in a failure of the mission. The objective of this research is to strengthen the framework behind safety in control and decision making, with an emphasis on utilizing the tools provided by numerical optimization, and on applications to problems in the space domain. The approach to safety is centered around Run Time Assurance (RTA), which relates to a control architecture that decouples the problem of enforcing safety constraints from the problem of optimizing performance objectives. Importantly, since an RTA approach is adopted, the results in this research are in general, complementary to those in performance-related research.

The first chapter presents a review on fundamental concepts related to optimization and safety-critical control. The second chapter provides a tutorial on RTA, and presents canonical approaches in this area. The remaining chapters address safety problems for three spacecraft systems: (i) Autonomous Rendezvous Proximity Operations and Docking (ARPOD) subject to collision avoidance constraints, (ii) attitude control subject to line-of-sight constraints, and (iii) utilizing planned collisions for safety and performance in free-flying spacecraft.

The contributions of this dissertation are described as follows. First, in Chapter 2, a study of RTA approaches is used to synthesize a novel RTA taxonomy. A set of canonical algorithms is identified, and various trade-offs are discussed among the approaches presented. Next, in Chapter 3, an optimal guidance and control framework is presented for safety-constrained optimal transfer trajectories to a safe set of closed natural motion trajectories under the Clohessy–Wiltshire-Hill dynamics. The planning problem is posed as a mixed integer program, which can be solved to global optimality under certain fuel- and power-minimizing cost functions. The algorithm is shown to be effective for enforcing safety in an RTA framework. In Chapter 4, a novel approach is presented for

enforcing safety constraints under nondeterministic dynamics, and this approach is applied to the problem of enforcing line-of-sight constraints for a torque-controlled spacecraft. The algorithm relies on computing reachable sets under a recovery maneuver online, and determining whether all realizations of the recovery maneuver safely reach an invariant backup set centered around a safe pointing direction. Finally, Chapter 5 introduces an approach to optimizing trajectories comprising planned collisions, and applies this to a three degree-of-freedom free-flying spacecraft. The planning algorithm integrates an empirically derived collision model into the constraints of a mixed integer program. Experiments comparing the efficiency of collision-free and collision-inclusive trajectories provide a proof-of-concept demonstration of the approach's capability to bring about practical performance enhancements. Moreover, a simulated case study shows the potential for application of the method as an online safety measure.

The research presents the possibility for many promising future directions. The theory of RTA may be generalized to consider more classification criteria, the effects of perception, reaction to component failures, or to system level concerns, such as the interaction between multiple layers. Furthermore, progress stands to be made on this front in terms of strengthening the theory for nondeterministic and stochastic systems. It would be interesting to consider a deeper analysis on the role of machine learning or artificial intelligence in the construction of backup controllers; *e.g.* it may be possible to use reinforcement learning for a backup controller that protects the system while it explores. In this research, safety filters are constructed for the translational and rotational states of a spacecraft separately. One next step would be to design and test an RTA mechanism for a combined 6 degree-of-freedom spacecraft model. The collision-inclusive planning may also be generalized to a 6 degree-of-freedom model, or applied to other systems, such as aerial robots. Additionally, there is much left to be considered in terms of designing practical damage-minimizing RTA systems. For example, the theory behind this may be extended to a broader class of systems, or to other applications such as road vehicles. One may also consider alternative planning strategies for this problem, or the use of offline evaluations of the planner to train a controller for this purpose.

REFERENCES

[1] R. P. Cohen, "Formal verification and validation of convex optimization algorithms for model predictive control," Ph.D. dissertation, Georgia Institute of Technology, 2018.

[2] R. Cohen, E. Feron, and P.-L. Garoche, "Verification and validation of convex optimization algorithms for model predictive control," *Journal of Aerospace Information Systems*, vol. 17, no. 5, pp. 257–270, 2020.

[3] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[4] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*, 1st. New York, NY, USA: Cambridge University Press, 2017, ISBN: 1107652871, 9781107652873.

[5] M. Lubin, I. Zadik, and J. P. Vielma, "Mixed-integer convex representability," in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2017, pp. 392–404.

[6] R. E. Davis, D. A. Kendrick, and M. Weitzman, "A branch-and-bound algorithm for zero-one mixed integer programming problems," *Operations Research*, vol. 19, no. 4, pp. 1036–1044, 1971.

[7] A. Del Pia, S. S. Dey, and M. Molinaro, "Mixed-integer quadratic programming is in np," *Mathematical Programming*, vol. 162, no. 1-2, pp. 225–240, 2017.

[8] A. Richards and J. How, "Mixed-integer programming for control," in *American Control Conference, 2005. Proceedings of the 2005*, IEEE, 2005, pp. 2676–2683.

[9] C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 115–140, 2019.

[10] I. I. Cplex, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[11] G. OPTIMIZATION, "Inc. gurobi optimizer reference manual, 2015," *URL: http://www. gurobi. com*, 2014.

[12] A. Mosek, "The mosek optimization software," *Online at http://www. mosek. com*, vol. 54, no. 2-1, p. 5, 2010.

[13] S. Karaman, R. G. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *2008 47th IEEE Conference on Decision and Control*, IEEE, 2008, pp. 2117–2122.

[14] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 81–87.

[15] R. Alur, *Principles of Cyber-Physical Systems*. Cambridge, MA: The MIT Press, 2015.

[16] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid dynamical systems*. Princeton University Press, 2012.

[17] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[18] N. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT press, 2011.

[19] J. D. Schierman, M. D. DeVore, N. D. Richards, N. Gandhi, J. K. Cooper, K. R. Horneman, S. Stoller, and S. Smolka, "Runtime assurance framework development for highly adaptive flight control systems," Barron Associates, Inc. Charlottesville, Tech. Rep., 2015.

[20] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre, *Viability theory: new directions*. Springer Science & Business Media, 2011.

[21] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.

[22] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, 2020.

[23] T. Gurriet, M. Mote, A. Singletary, A. D. Ames, and E. Feron, "Scalable controlled set invariance framework with practical safety guarantees. in decision and control," in *2019 IEEE Conference on Decision and Control (CDC)*, IEEE, 2019.

[24] D. Mayne, "An apologia for stabilising terminal conditions in model predictive control," *International Journal of Control*, vol. 86, no. 11, pp. 2090–2095, 2013.

[25] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[26] B. T. Lopez and J. P. How, "Aggressive 3-d collision avoidance for high-speed navigation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5759–5765.

[27] O. Sanni, M. Mote, D. Delahaye, M. Gariel, T. Khamvilai, E. Feron, and S. Saber, *Ariadne: A common-sense thread for enabling provable safety in air mobility systems with unreliable components*, 2021.

[28] T. Schouwenaars, M. Valenti, E. Feron, and J. How, "Implementation and flight test results of milp-based uav guidance," in *2005 IEEE Aerospace Conference*, IEEE, 2005, pp. 1–13.

[29] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of the 2004 American control conference*, IEEE, vol. 6, 2004, pp. 5576–5581.

[30] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Natural motion-based trajectories for automatic spacecraft proximity operation collision avoidance," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[31] W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[32] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*, Springer, 2017, pp. 97–117.

[33] H.-D. Tran, X. Yang, D. M. Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, and T. T. Johnson, "Nnv: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems," in *International Conference on Computer Aided Verification*, Springer, 2020, pp. 3–17.

[34] S. Bak, H.-D. Tran, K. Hobbs, and T. T. Johnson, "Improved geometric path enumeration for verifying relu neural networks," in *International Conference on Computer Aided Verification*, Springer, 2020, pp. 66–96.

[35] S. Gokulanathan, A. Feldsher, A. Malca, C. Barrett, and G. Katz, "Simplifying neural networks using formal verification," in *NASA Formal Methods Symposium*, Springer, 2020, pp. 85–93.

[36] J. C. Knight, "Safety critical systems: Challenges and directions," in *Proceedings of the 24th International Conference on Software Engineering*, 2002, pp. 547–550.

[37] M. Mote, C. W. Hays, A. R. Collins, E. Feron, and K. L. Hobbs, "Natural motion-based trajectories for automatic spacecraft collision avoidance during proximity operations," in *IEEE Aerospace*, 2021.

[38] S. Bak, K. Manamcheri, S. Mitra, and M. Caccamo, "Sandboxing controllers for cyber-physical systems," in *International Conference on Cyber-physical systems*, ser. ICCPS, 2011.

[39] D. E. Swihart, A. F. Barfield, E. M. Griffin, R. C. Lehmann, S. C. Whitcomb, B. Flynn, M. A. Skoog, and K. E. Prosser, "Automatic ground collision avoidance system design, integration, & flight test," *IEEE Aerospace and Electronic Systems Magazine*, vol. 26, no. 5, pp. 4–11, 2011.

[40] E. M. Griffin, R. M. Turner, S. C. Whitcomb, D. E. Swihart, J. M. Bier, K. L. Hobbs, and A. C. Burns, "Automatic ground collision avoidance system design for pre-block 40 f-16 configurations," in *Asia-Pacific International Symposium on Aerospace Technology*, 2012.

[41] M. Aiello, J. Berryman, J. Grohs, and J. Schierman, "Run-time assurance for advanced flight-critical control systems," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 8041.

[42] A. Burns, K. Hobbs, J. Bier, and S. Whitcomb, "Advanced capabilities for the analog flight control f-16," in *NATO Sensors and Electronics Technology Panel 168 Symposium*, 2012.

[43] N. Minoiu Enache, S. Mammar, M. Netto, and B. Lusetti, "Driver steering assistance for lane-departure avoidance based on hybrid automata and composite lyapunov function," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 28–39, 2010.

[44] K. L. Hobbs, "Elicitation and formal specification of run time assurance requirements for aerospace collision avoidance systems," Ph.D. dissertation, Georgia Institute of Technology, 2020.

[45] M. Abate, E. Feron, and S. Coogan, "Monitor-based runtime assurance for temporal logic specifications," *arXiv preprint arXiv:1908.03284*, 2019.

[46] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.

[47] C. J. Tomlin, J. Lygeros, and S. S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.

[48] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution manoeuvres," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 110–120, 2001.

[49] S. Liu, M. Watterson, S. Tang, and V. Kumar, "High speed navigation for quadrotors with limited onboard sensing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1484–1491.

[50] J. H. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 335–354, 2011.

[51] G. M. Hoffmann and C. J. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles," in *2008 47th IEEE Conference on Decision and Control*, IEEE, 2008, pp. 4357–4363.

[52] C.-F. Lin, J.-C. Juang, and K.-R. Li, "Active collision avoidance system for steering control of autonomous vehicles," *IET Intelligent Transport Systems*, vol. 8, no. 6, pp. 550–557, 2014.

[53] V. Muthukumaran, R. G. Sanfelice, and G. H. Elkaim, "A hybrid control strategy for autonomous navigation while avoiding multiple obstacles at unknown locations," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2019, pp. 1042–1047.

[54] D. Phan, J. Yang, R. Grosu, S. A. Smolka, and S. D. Stoller, "Collision avoidance for mobile robots with limited sensing and limited information about moving obstacles," *Formal Methods in System Design*, vol. 51, no. 1, pp. 62–86, 2017.

[55] T. Gurriet, M. Tucker, A. Duburcq, G. Boeris, and A. D. Ames, "Towards variable assistance for lower body exoskeletons," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 266–273, 2019.

[56] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamical bipedal walking with precise footstep placement," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 147–154, 2015.

[57] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 827–834.

[58] A. Singletary, P. Nilsson, T. Gurriet, and A. D. Ames, "Online active safety for robotic manipulators," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 173–178.

[59] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2017.

[60] C. Hu, Z. Wang, Y. Qin, Y. Huang, J. Wang, and R. Wang, "Lane keeping control of autonomous vehicles with prescribed performance considering the rollover prevention and input saturation," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[61] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2018, pp. 98–106.

[62] A. Singletary, T. Gurriet, P. Nilsson, and A. D. Ames, "Safety-critical rapid aerial exploration of unknown environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 270–10 276.

[63] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Transactions on Control Systems Technology*, 2019.

[64] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[65] D. Pickem, L. Wang, P. Glotfelter, Y. Diaz-Mercado, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "Safe, remote-access swarm robotics research on the robotarium," *arXiv preprint arXiv:1604.00640*, 2016.

[66] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1699–1706.

[67] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[68] S. Wilson, P. Glotfelter, S. Mayya, G. Notomista, Y. Emam, X. Cai, and M. Egerstedt, "The robotarium: Automation of a remotely accessible, multi-robot testbed," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2922–2929, 2021.

[69] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," *arXiv preprint arXiv:2104.11332*, 2021.

[70] A. Isaly, B. C. Allen, R. G. Sanfelice, and W. E. Dixon, "Zeroing control barrier functions for safe volitional pedaling in a motorized cycle," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 218–223, 2020.

[71] L. Sha, R. Rajkumar, and M. Gagliardl, "A software architecture for dependable and evolvable industrial computing systems.," Carnegie-Mellon University Software Engineering Institute, Pittsburgh, PA, Tech. Rep., 1995.

[72] J. G. Rivera, A. A. Danylyszyn, C. B. Weinstock, L. R. Sha, and M. J. Gagliardi, "An architectural description of the simplex architecture.," Carnegie-Mellon University Software Engineering Institute, Pittsburgh, PA, Tech. Rep., 1996.

[73] C. Reis, A. Barth, and C. Pizano, "Browser security: Lessons from google chrome," *Commun. ACM.*, vol. 52, pp. 45–49, 2009.

[74] U. Mehmood, S. Bak, S. A. Smolka, and S. D. Stoller, "Safe cps from unsafe controllers," *arXiv preprint arXiv:2102.12981*, 2021.

[75] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 3420–3431.

[76] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187 249–187 275, 2020.

[77] T. Gurriet, "Applied safety critical control," Ph.D. dissertation, California Institute of Technology, 2020.

[78] J. Aubin, *Viability theory. modern birkhäuser classics*, 2009.

[79] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[80] J. H. Gillula, S. Kaynama, and C. J. Tomlin, "Sampling-based approximation of the viability kernel for high-dimensional linear sampled-data systems," in *Proceedings of the 17th international conference on Hybrid systems: computation and control*, 2014, pp. 173–182.

[81] I. Mitchell, "A summary of recent progress on efficient parametric approximations of viability and discriminating kernels.," in *SNR@ CAV*, 2015, pp. 23–31.

[82] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[83] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[84] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2004, pp. 477–492.

[85] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[86] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 6271–6278.

[87] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.

[88] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation.," in *Robotics: Science and Systems*, 2017.

[89] C. Santoyo, M. Dutreix, and S. Coogan, "A barrier function approach to finite-time stochastic system verification and control," *Automatica*, vol. 125, p. 109 439, 2021.

[90] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 1991–1996.

[91] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 322–328.

[92] H. Seywald and R. Kumar, "Desensitized optimal trajectories," *Analytical Mechanics Associates Rept*, pp. 03–16, 2003.

[93] M. A. Patterson and A. V. Rao, "Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 1, pp. 1–37, 2014.

[94] M. Mote, M. Egerstedt, E. Feron, A. Bylard, and M. Pavone, "Collision-inclusive trajectory optimization for free-flying spacecraft," *Journal of Guidance, Control, and Dynamics*, pp. 1–12, 2020.

[95] B. Evans, "Dawn of a new era," in *Partnership in Space*, Springer, 2014, pp. 453–482.

[96] D. Holland, "6.4. 1 a case study of the near-catastrophic mir-progress 234 collision with emphasis on the human factors/systems-level issues surrounding this mishap," in *INCOSE International Symposium*, Wiley Online Library, vol. 12, 2002, pp. 820–827.

[97] J. Oberg, "Shuttle-mir's lessons for the international space station," *IEEE Spectrum*, vol. 35, no. 6, pp. 28–37, 1998.

[98] L. D. thomas, "Selected systems engineering process deificiencies and their consequences," National Aeronautics and Space Administration George C. Marshall Space Flight Center, Tech. Rep., 2007.

[99] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *2001 European control conference (ECC)*, IEEE, 2001, pp. 2603–2608.

[100] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002.

[101] A. S. LeValley, "A mixed integer programming framework for the fuel optimal guidance of complex spacecraft rendezvous and proximity operation missions," Air Force Institute of Technology, Wright-Patterson AFB, OH, Tech. Rep., 2019.

[102] M. Tam and E. G. Lightsey, "Constrained spacecraft reorientation using mixed integer convex programming," *Acta Astronautica*, vol. 127, pp. 31–40, 2016.

[103] B. Kelly and S. De Picciotto, "Probability based optimal collision avoidance maneuvers," in *Space 2005*, 2005, p. 6775.

[104] C. Bombardelli and J. Hernando-Ayuso, "Optimal impulsive collision avoidance in low earth orbit," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 217–225, 2015.

[105] D. Mishne and E. Edlerman, "Collision-avoidance maneuver of satellites using drag and solar radiation pressure," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 5, pp. 1191–1205, 2017.

[106] C. Petersen, A. Jaunzemis, M. Baldwin, M. Holzinger, and I. Kolmanovsky, "Model predictive control and extended command governor for improving robustness of relative motion guidance and control," *Advances In The Astronautical Sciences*, vol. 152, pp. 701–718, 2014.

[107] A. Weiss, C. Petersen, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Safe positively invariant sets for spacecraft obstacle avoidance," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 720–732, 2014.

[108] K. Lesser, M. Oishi, and R. S. Erwin, "Stochastic reachability for control of spacecraft relative motion," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, IEEE, 2013, pp. 4705–4712.

[109] A. Nemirovski and A. Shapiro, "Scenario approximations of chance constraints," in *Probabilistic and randomized methods for design under uncertainty*, Springer, 2006, pp. 3–47.

[110] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[111] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5876.

[112] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2152–2159.

[113] G. R. Frey, C. D. Petersen, F. A. Leve, I. V. Kolmanovsky, and A. R. Girard, "Constrained spacecraft relative motion planning exploiting periodic natural motion trajectories and invariance," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 12, pp. 3100–3115, 2017.

[114] G. R. Frey, C. D. Petersen, F. A. Leve, A. R. Girard, and I. V. Kolmanovsky, "Invariance-based spacecraft relative motion planning incorporating bounded disturbances and minimum thrust constraints," in *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 658–663.

[115] G. R. Frey, C. D. Petersen, F. A. Leve, I. V. Kolmanovsky, and A. R. Girard, "Incorporating periodic and non-periodic natural motion trajectories into constrained invariance-based spacecraft relative motion planning," in *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, IEEE, 2017, pp. 1811–1816.

[116] G. R. Frey, C. D. Petersen, F. A. Leve, E. Garone, I. V. Kolmanovsky, and A. R. Girard, "Time shift governor for coordinated control of two spacecraft formations," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 296–301, 2016.

[117] G. W. Hill, "Researches in the lunar theory," *American journal of Mathematics*, vol. 1, no. 1, pp. 5–26, 1878.

[118] F. S. Hillier, G. J. Lieberman, F. Hillier, and G. Lieberman, *Introduction to Operations Research*. McGraw-Hill, New York, USA, 2004.

[119] M Guelman and M Aleshin, "Optimal bounded low-thrust rendezvous with fixed terminal-approach direction," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 378–385, 2001.

[120] C. J. Pardis and T. E. Carter, "Optimal power-limited rendezvous with thrust saturation," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, pp. 1145–1150, 1995.

[121] C. Llanes, M. Abate, and S. Coogan, "Safety from in-the-loop reachability for cyber-physical systems," in *2021 Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems (CAADCPS)*, CPS-IoT week, 2021.

[122] M. Abate and S. Coogan, "Enforcing safety at runtime for systems with disturbances," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 2038–2043.

[123] M. Abate, M. Mote, E. Feron, and S. Coogan, "Verification and runtime assurance for dynamical systems with uncertainty," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '21, Nashville, Tennessee: Association for Computing Machinery, 2021, ISBN: 9781450383394.

[124] S. Coogan, "Mixed monotonicity for reachability and safety in dynamical systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 5074–5085.

[125] J. Gouzé and K. Hadeler, "Monotone flows and order intervals," *Nonlinear World*, vol. 1, pp. 23–34, 1994.

[126] D. Angeli, G. A. Enciso, and E. D. Sontag, "A small-gain result for orthant-monotone systems under mixed feedback," *Systems & Control Letters*, vol. 68, pp. 9 –19, 2014.

[127] G. Enciso, H. Smith, and E. Sontag, "Nonmonotone systems decomposable into monotone systems with negative feedback," *Journal of Differential Equations*, vol. 224, no. 1, pp. 205 –227, 2006.

[128] L. Yang, O. Mickelin, and N. Ozay, "On sufficient conditions for mixed monotonicity," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5080–5085, 2019.

[129] S. Coogan and M. Arcak, "Stability of traffic flow networks with a polytree topology," *Automatica*, vol. 66, no. C, pp. 246–253, Apr. 2016.

[130] H. L. Smith, "The discrete dynamics of monotonically decomposable maps," *Journal of Mathematical Biology*, vol. 53, no. 4, p. 747, 2006.

[131] S. Coogan and M. Arcak, "Efficient finite abstraction of mixed monotone systems," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, 2015, pp. 58–67.

[132] P.-J. Meyer, A. Devonport, and M. Arcak, "Tira: Toolbox for interval reachability analysis," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '19, Montreal, Quebec, Canada: Association for Computing Machinery, 2019, 224–229, ISBN: 9781450362825.

[133] P. Meyer and D. V. Dimarogonas, "Hierarchical decomposition of LTL synthesis problem for nonlinear control systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4676–4683, 2019.

[134] M. Abate and S. Coogan, "Computing robustly forward invariant sets for mixed-monotone systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 4553–4559.

[135] M. Abate, M. Dutreix, and S. Coogan, "Tight decomposition functions for continuous-time mixed-monotone systems with disturbances," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 139–144, 2021.

[136] M. Khajenejad and S. Z. Yong, *Tight remainder-form decomposition functions with applications to constrained reachability and interval observer design*, 2021. arXiv: 2103.08638 [math.OC].

[137] M. Abate and S. Coogan, "Improving the fidelity of mixed-monotone reachable set approximations via state transformations," in *2021 American Control Conference*, 2021.

[138] L. Pedersen, D. Kortenkamp, D. Wettergreen, I Nourbakhsh, and D. Korsmeyer, "A survey of space robotics," *7th International Symposium on Artificial, Intelligence, Robotics, and Automation in Space*, 2003.

[139] S. E. Fredrickson, L. W. Abbott, S. Duran, J. D. Jochim, J. W. Studak, J. D. Wagenknecht, and N. M. Williams, "Mini aercam: Development of a free-flying nanosatellite inspection robot," in *Space Systems Technology and Operations*, International Society for Optics and Photonics, vol. 5088, 2003, pp. 97–112.

[140] T. Williams and S. Tanygin, "On-orbit engineering tests of the aercam sprint robotic camera vehicle," *Spaceflight mechanics 1998*, pp. 1001–1020, 1998.

[141] I. Boyd, R. Buenconsejo, D Piskorz, B Lal, K. Crane, and E De La Rosa Blanco, "On-orbit manufacturing and assembly of spacecraft," *IDA Paper P-8335, Institute for Defense Analysis, Alexandria, VA*, 2017.

[142] A. Bylard, R. MacPherson, B. Hockman, M. R. Cutkosky, and M. Pavone, "Robust capture and deorbit of rocket body debris using controllable dry adhesion," in *Aerospace Conference, 2017 IEEE*, IEEE, 2017, pp. 1–9.

[143] P. Acquatella, "Development of automation & robotics in space exploration," in *Proceedings of the AIAA SPACE 2009 Conference & Exposition. California*, Citeseer, 2009, pp. 1–7.

[144] M. Bualat, J. Barlow, T. Fong, C. Provencher, and T. Smith, "Astrobee: Developing a free-flying robot for the international space station," *AIAA SPACE 2015 Conference and Exposition*, p. 4643, 2015.

[145] D. Miller, A Saenz-Otero, J Wertz, A Chen, G Berkowski, C Brodel, S Carlson, D Carpenter, S Chen, S Cheng, *et al.*, "Spheres: A testbed for long duration satellite formation flying in micro-gravity conditions," in *Proceedings of the AAS/AIAA space flight mechanics meeting*, Clearwater, Florida, January, 2000, pp. 167–179.

[146] C. Jewison and D. W. Miller, "Probabilistic trajectory optimization under uncertain path constraints for close proximity operations," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 9, pp. 1843–1858, 2018.

[147] G. E. Chamitoff, A. Saenz-Otero, J. G. Katz, S. Ulrich, B. J. Morrell, and P. W. Gibbens, "Real-time maneuver optimization of space-based robots in a dynamic environment: Theory and on-orbit experiments," *Acta Astronautica*, vol. 142, pp. 170–183, 2018.

[148] B. Morrell, "Enhancing 3d autonomous navigation through obstacle fields: Homogeneous localisation and mapping, with obstacle-aware trajectory optimisation," 2018.

[149] F. Rehnmark, W. Bluethmann, J. Mehling, R. O. Ambrose, M. Diftler, M. Chu, and R. Necessary, "Robonaut: The 'short list' of technology hurdles," *Computer*, vol. 38, no. 1, pp. 28–37, 2005.

[150] J. Virgili-Llop and M. Romano, "Astrobatics: Demonstrating propellantless robotic maneuvering onboard the international space station," Naval Postgraduate School Monterey United States, Tech. Rep., 2018.

[151] K. P. Alsup, "Robotic spacecraft hopping: Application and analysis," Naval Postgraduate School Monterey United States, Tech. Rep., 2018.

[152] W. Xi and C. D. Remy, "Optimal gaits and motions for legged robots," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, IEEE, 2014, pp. 3259–3265.

[153] F. Giardina and F. Iida, "Efficient and stable locomotion for impulse-actuated robots using strictly convex foot shapes," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 674–685, 2018.

[154] D. Serra, A. C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, "An optimal trajectory planner for a robotic batting task: The table tennis example.," in *ICINCO (2)*, 2016, pp. 90–101.

[155] M. W. Spong, "Impact controllability of an air hockey puck," *Systems & Control Letters*, vol. 42, no. 5, pp. 333–345, 2001.

[156] W. Xin, J. Hourdos, P. Michalopoulos, and G. Davis, "The less-than-perfect driver: A model of collision-inclusive car-following behavior," *Transportation research record*, vol. 2088, no. 1, pp. 126–137, 2008.

[157] J Kiefer, M Ward, and M Costello, "Rotorcraft hard landing mitigation using robotic landing gear," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 3, p. 031 003, 2016.

[158] S. H. Lee, B.-J. Yi, S. H. Kim, and Y. K. Kwak, "Analysis on impact propagation of docking platform for spacecraft," in *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA.*, IEEE, vol. 1, 2001, pp. 413–420.

[159] M. A. Estrada, B. Hockman, A. Bylard, E. W. Hawkes, M. R. Cutkosky, and M. Pavone, "Free-flyer acquisition of spinning objects with gecko-inspired adhesives," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 4907–4913.

[160] B. J. Hockman, A. Frick, R. G. Reid, I. A. Nesnas, and M. Pavone, "Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies," *Journal of Field Robotics*, vol. 34, no. 1, pp. 5–24, 2017.

[161] M. Posa, M. Tobenkin, and R. Tedrake, "Stability analysis and control of rigid-body systems with impacts and friction," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1423–1437, 2016.

[162] Y. Mulgaonkar, A. Makineni, L. Guerrero-Bonilla, and V. Kumar, "Robust aerial robot swarms without collision avoidance," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 596–603, 2018.

[163] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Localization in densely packed swarms using interrobot collisions as a sensing modality," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 21–34, 2019.

[164] S. Connell, "Crazy cat [video]," 2010.

[165] M. Mote, J. P. Afman, and E. Feron, "Robotic trajectory planning through collisional interaction," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, IEEE, 2017, pp. 1144–1149.

[166] D Seto, B. H. Krogh, L. Sha, and A Chutinan, "Dynamic control system upgrade using the simplex architecture," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 72–80, 1998.

[167] E. C. Kerrigan and J. M. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," in *Proc. UKACC International Conference*, 2000.

[168] M. Morari, "Hybrid system analysis and control via mixed integer optimization," *IFAC Proceedings Volumes*, vol. 34, no. 25, pp. 1–12, 2001.

[169] A. Chatterjee, *Rigid body collisions: some general considerations, new collision laws, and some experimental data*. Cornell University Ithaca, NY, 1997.

[170] E. Cataldo and R. Sampaio, "A brief review and a new treatment for rigid bodies collision models," *Journal of the Brazilian Society of Mechanical Sciences*, vol. 23, no. 1, pp. 63–78, 2001.

[171] S. A. Schneider and R. H. Cannon, "Object impedance control for cooperative manipulation: Theory and experimental results," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.

[172] M. Mote, M. Egerstedt, A. Bylard, E. Feron, and M. Pavone, "Collision-Inclusive Trajectory Optimization for Spacecraft - Experimental Comparison," Jan. 2020.

[173] M. Mote, E. Feron, A. Bylard, M. Egerstedt, and M. Pavone, "Online Safety through Utilization of Collisions," Jan. 2020.

[174] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 3592–3599.

[175] J. A. Starek, E. Schmerling, G. D. Maher, B. W. Barbee, and M. Pavone, "Fast, safe, propellant-efficient spacecraft motion planning under clohessy–wiltshire–hill dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 418–438, 2017.